NASA CONTRACTOR
REPORT

NASA CR-150022

(NASA-CR-150022) U1108 PERFORMANCE MODEL (SDC/Integrated Services, Inc.) 63 p HC A04/MF A01 CSCL 09B

N77-10808

Unclas
G3/61 07991

U1108 PERFORMANCE MODEL

By G. S. Trachta
SDC/Integrated Services, Inc.
Slidell Computer Complex
1010 Gause Blvd.
Slidell, Louisiana 70458

June 17, 1976

Prepared for

NASA-GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO. NASA CR-150022	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
4 TITLE AND SUBTITLE U1108 Performance Model		5. REPORT DATE June 17, 1976 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) G. Trachta		8. PERFORMING ORGANIZATION REPORT # SDC/ISI 236/001
9. PERFORMING ORGANIZATION NAME AND ADDRESS SDC/Integrated Services, Inc. Slidell Computer Complex Slidell, Louisiana 70458		10. WORK UNIT NO. 11. CONTRACT OR GRANT NO. NAS8-31642 13. TYPE OF REPORT & PERIOD COVERED
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D. C. 20546		Contractor 14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

This work was sponsored by the NASA-Marshall Space Flight Center, Alabama, under Contract NAS8-31642.

16. ABSTRACT

A model of Univac 1108 work flow has been developed to assist in performance evaluation studies and configuration planning. Workload profiles and system configurations are parameterized for ease of experimental modification. Outputs include capacity estimates and performance evaluation functions. The U1108 system is conceptualized as a service network; classifcal queueing theory is used to evaluate network dynamics.

17. KEY WORDS	18. DISTRIBUTION STATEMENT Unclassified-Unlimited J. T. Shepherd Director, Admin and Prog Support

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	62	NTIS

MSFC - Form 3292 (Rev December 1972)

For sale by National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

1.0 INTRODUCTION

It has become an increasingly emphasized desire of the managers of large scale computer centers to make objective, verifiable statements about computer performance and capacity. This desire has become more urgent as it has become more difficult of achieving. The complexity of operation that has made the intuitive concepts of computer performance unreliable has made the previously parttime art of computer evaluation a specialized discipline.

In previous generations of computers, prior to processing multiple runs simultaneously and configuring central processing units and peripherals with plug-in flexibility, performance evaluation was a simple consideration of runs processed per unit time. Sophisticates of the art dealt with CPU time and some sources of delay. To configure a system to a workload one considered average instructions, amounts of data, processor cycle times and output speeds. All tasks were processed serially, one after the other, and system impact was calculated by summing up the component times of a few prototype jobs. Systems were tuned by watching them run.

Performance evaluation in the multiprogramming/multiprocessing generation is utterly transformed. At any moment, numerous runs are active within the computer, competing for services from all system components. The same run may compete simultaneously for different computer services. The impact of a run on system performance is a function of the total workload during the life of the run. The history of a program's activity in the computer system is never exactly the same for any two executions.

The Slidell Computer Complex (SCC) operates Univac 1108 computer systems in support of batch and terminal applications. User requirements vary widely in terms of program size, processor requirements and mass storage usage. The environment is in every way typical of a large scale, open shop computer facility.

The SCC conducts an ongoing analysis of U1108 work flow to establish capacity estimates and to measure performance. A major goal has been to define the capacity function in terms of two independent classes of variables - computer configuration and workload profile. It is recognized that variations in system performance result from changes in both the physical structure of the machine and the requirements structure of the workload.

A number of approaches to performance evaluation have been considered at the SCC. Attaching electronic probe monitors to various critical system components is being considered. System performance has been monitored by a special software implementation (Software Instrumentation Program - SIP). Regression analysis has been used to find linear relationships between CPU accumulations and selected measureable parameters. Reasonable capacity estimates have been obtained from regression analysis but the equations are difficult to adjust for changing environments. It is not always apparent how the so-called independent variables respond to drastic shifts in workload and configuration. This shortcoming is fundamental. The relationship between meaningful independent variables and system performance is not expressable as a regression curve. Trend analysis fails when the trend changes.

The SCC's most recent performance evaluation tool, a U1108 performance model, considers the computer to be a network of service centers. The workload is conceived as a set of service requests. Each request is queued and processed under control of user programs and system software. Capacity is defined as the work level at which the network saturates. The configuration and workload are defined

in terms of independent, predictable parameters. Queueing theory is used to calculate the work flow dynamics. Section 2.0 describes a brief, intuitive development of the theory. Section 3.0 describes the model. Section 4.0 is a detailed development of the numeric techniques used in the model. An example of model application is presented in Section 5.0. Section 6.0 is a user's guide to the computer program implementing the model and Section 7.0 presents the program listing.

2.0 SERVICE QUEUES

If a service center is busy at the time a request for services arrives, a wait period (or queue time) accrues. The average queue time for a series of requests can be estimated by queueing theory.

Consider a service center as depicted below:

Each service request has two attributes that determine its interaction with the service center: its arrival time and the amount of service requested. The service center's performance is determined by the number of servers (the number of simultaneous requests that it can serve) and the processing rate of each server. Estimation of these parameters allows calculation of the probability of an arrival in an arbitrary time period and the probability of all servers being busy at the time of an arrival. The probability of an arbitrary wait period may then be expressed and integrated with respect to time to yield the average wait time.

To estimate the probability of an arrival in an arbitrary interval of time, two assumptions are made to simplify the calculations:

i. The probability of an arrival in t seconds is proporational to t (i.e. the longer the wait for a service request, the greater the chances of receiving one).

ii. The probability of more than one arrival in t seconds shrinks faster than t (i.e. arrivals are sequential and not clustered).

These assumptions allow the probability of arrival to be expressed by the Poisson distribution:

$$P \text{ (n arrivals in time t)} = \frac{(at)^n}{n!} e^{-at}$$

where a is the average arrival rate.

NOTE: The notation P(X) will be used to denote "the probability of event X".

Similar considerations lead to an exponetial representation of the service rate.

P (n requests serviced in time t) $= e^{bt}$

where b is the average service rate.

Using these probability distributions, we can express the average queue time in terms of

i. the average arrival rate,
ii. the average service rate, and
iii. the number of servers.

For the U1108 performance model, the number of servers is a computer configuration parameter. The average service rate is a function of workload and configuration. The average arrival rate may be considered an independent variable in the queue calculation; for a given arrival rate, a determinable queue time results.

If we assume that queued results are processed on a first-come first-served basis and that requests do not defect from the queue before being served, then a simple queue time calculation can be formulated. The derivation involves development of differential equations for two cases.

case 1. There is no arrival in an arbitrarily small period of time.
case 2. There is exactly one arrival in an arbitrarily small period.

With the assumption of Poisson arrivals, these two cases are the only two possible since the arrivals do not cluster. Average queue time can be expressed as:

$$\text{QUEUE }(A,B,C) = \left(\frac{1}{BC-A}\right)\left(\frac{P^{C}C}{C!\,(C-P)}\right)\left[\left(\frac{P^{C}C}{C!\,(D-P)}\right)\sum_{i=1}^{C}\left(\frac{P^{C-i}}{(C-i)!}\right)\right]^{-1} \quad \text{if, and only if, } BC > A$$

where A = average arrival rate
B = average service rate
C = number of servers
P = A/B

It should be noted that if A is greater than or equal to BC, the average queue time is infinite and the service center is saturated. That is, if the arrival rate exceeds the product of the service rate and the number of servers, the service center is overloaded. Capacity is conceived as the upper limit of arrival rates that do not exceed the service rate times the number of servers. Within a network of service centers, the capacity for the network is the lowest input rate which saturates one of the centers.

3.0 WORKFLOW MODEL

To model the U1108 workflow, we wish to know what happens to a computer task (run) during its active life in the computer. We know that part of this time is spent in the service queues. Other delays occur that are related to the structure of the run and the state of the computer system.

We may categorize this elapsed time as:

i. service time,
ii. service queue time,
iii. memory queue time,
u. voluntary delay time, and
v. involuntary delay time.

Service time includes the CPU time and the I/O traffic time. CPU time is a function of the instruction sequence of the run and the CPU/main memory cycle speed.

I/O traffic time is a function of data words transferred, record size, and the speed of the I/O device. Since a given run may have its I/O requirements serviced by a variety of devices, each with its own speed, the service time is dependent on the probability of using a specific I/O device. These probabilities will be called the I/O traffic patterns.

Service queue time is the wait period for CPU and I/O traffic services.

Memory queue time is the wait period prior to receiving an allocation of main memory. This allocation must be long enough to encompass both the service and service queue times.

Voluntary delay time includes periods when the run is temporarily requesting no services. Such delays typically occur on interactive runs input from demand terminals when the user is not transmitting requests.

Involuntary delay time consists of periods when the run is prevented from making service requests. The usual cause is a request for I/O from a magnetic tape servo before a tape has been physically mounted.

Runs, of course, do not accumulate elapsed time as might be implied by this categorization, getting all the service queue time, then all the service time, then all voluntary delay and so forth. The actual history of a run may involve many small increments of time in all of these categories. This organization of the elapsed time is important because it suggests a way to estimate it, not because it depicts a micro view of the life of a run.

To calculate queue times we consider the U1108 computer to be a network of service centers. The network contemplates three major computer services viz central processor (CPU) service, I/O traffic service and main memory service. It assumes that a task is main memory resident during the time it is queued for and receiving CPU and I/O services. The I/O traffic services are categorized by specific I/O device.

Figure 1 is a general schematic of the first part of the queueing network. As depicted, each I/O device (excluding unit record devices) is contemplated separately.

CPU and I/O requests flow to their respective service centers. The rate at which these services are requested, together with the rate at which CPU and I/O queue time are accumulated, make up the memory service input rate. The schematic seems to turn the actual operation of the computer inside out. Runs actually receive main memory allocation before CPU and I/O services. However, to calculate the main memory queue time, it is necessary first to calculate the CPU and I/O queues since this wait time is part of the main memory service request rate.

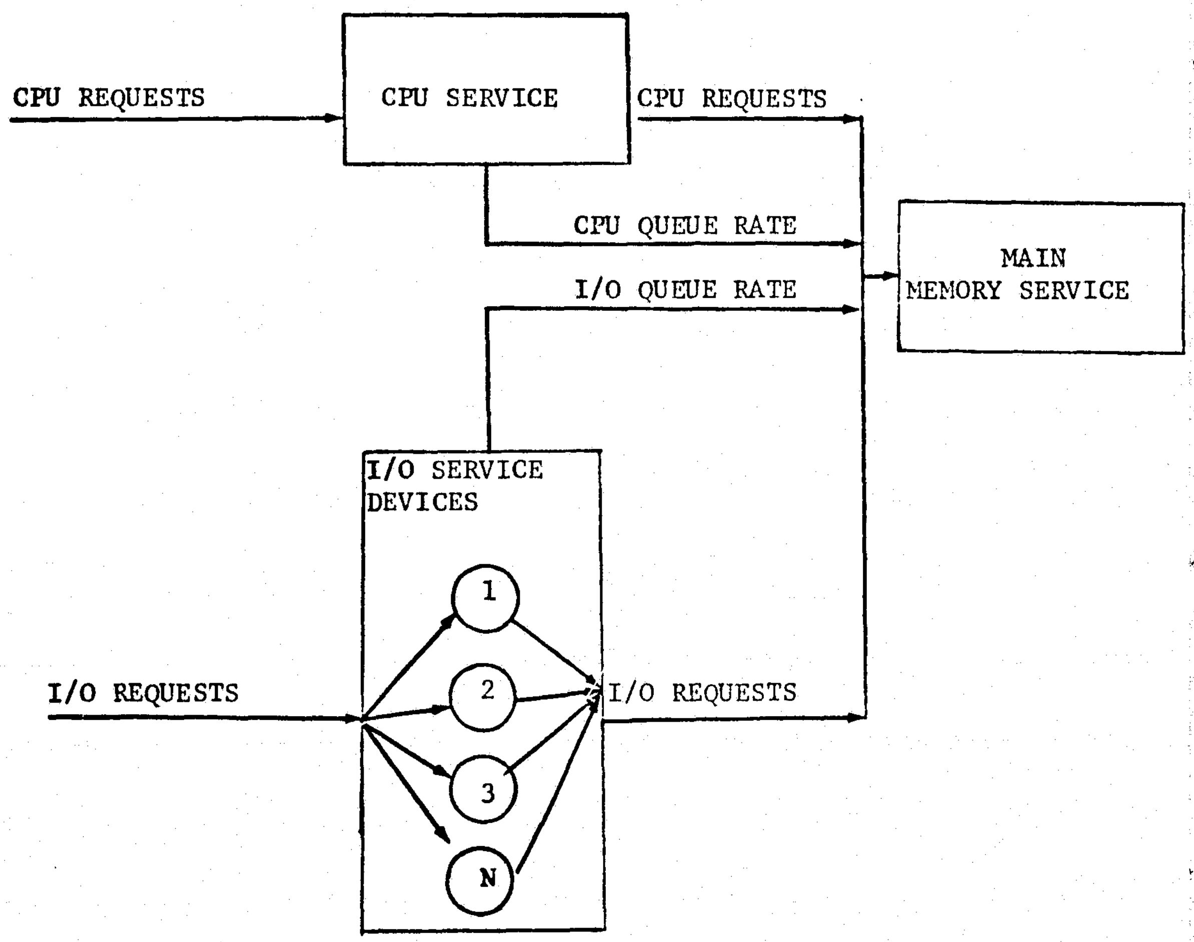

FIGURE 1.
QUEUEING NETWORK

The model also includes estimates of voluntary and involuntary delay time. These estimates plus service requests and queue times provide an average elapsed time estimate for a given work input rate.

As depicted in Figure 2, this estimate of the elapsed time rate is used as input to the batch delay service center. This center simulates the operator's control over batch runs. A software valve controlled by a console keyin prevents more than a specified number of batch runs becoming active at the same time. The batch delay queue estimates this unrecorded elapsed time and adjustments are made to the elapsed time estimate.

4.0 MODEL MATHEMATICS

The mathematics used in the model assume that the work input rate, the computer configuration and the workload profile are given. Performance parameters are computed from these three major variables.

4.1 WORKLOAD INPUT RATE

The operating system of the U1108 computer calculates an estimate of service requirements called the Standard Unit of Processing (SUP). The SUP accumulates the CPU time used by a run and estimates the I/O time. Taken collectively for all runs processed in a unit period of time, the SUP provides an estimate of the total service requirements.

The accuracy of the SUP estimate is variable. CPU time is taken from the internal clock and is an accurate measure of the requirements of a run except that all functions of the operating system are not included. The I/O time is estimated, based on words transferred, average access time and transfer times. The estimate assumes that I/O occurs on the mass storage device requested by the run even though another physical device may have been substituted by the operating system. The CPU and I/O time used to perform executive requests and execute control card functions are estimated from a table of fixed charges. The accuracy of these fixed charges may vary from run to run and it is also not apparent how much of the charge represents CPU time and how much I/O time.

These accuracy problems not withstanding the SUP is the best available estimate of collective service requirements. Benchmark runs indicate that it is accurate enough.

It is used by the model as the basic measure of performance. The computer input rate is expressed in terms of SUP hours per hour of effective computer time.

Effective computer time is defined as the time the computer produces output. It excludes downtime, idle time and the apparently productive time spent on runs which are active and, therefore, lost when a system failure occurs.

4.2 WORKLOAD PROFILE

The workload is profiled in terms of its impact on each element of the model. Specifically, the workload profile includes the following:

1. R_C = the rate of CPU requirements expressed as CPU time per SUP.

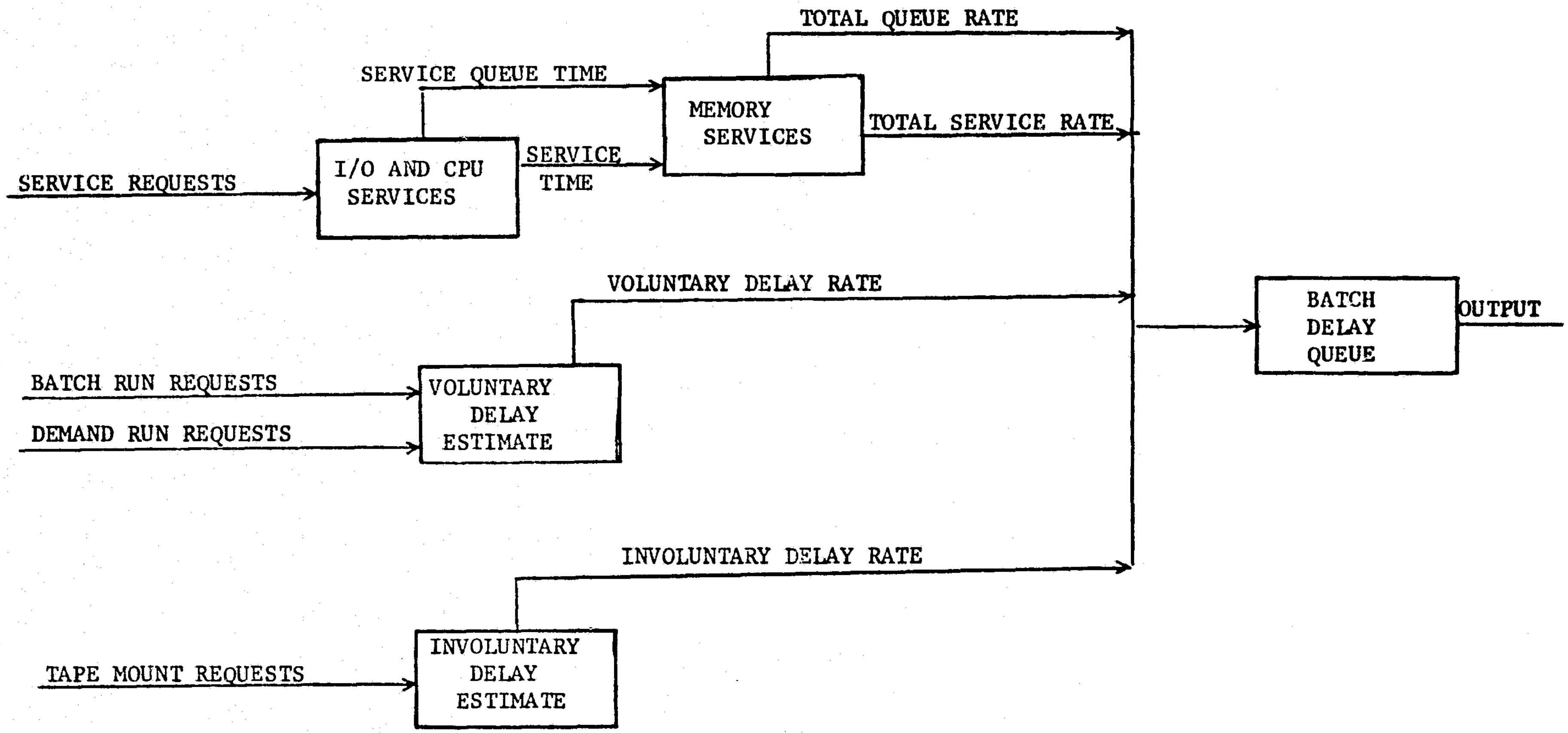

FIGURE 2. MODEL SCHEMATIC

2. R_I = the rate of I/O requirements expressed as words transferred per SUP.

3. $P(n)$ = the probability a given I/O request occurs on device n.

4. $\overline{W}(n)$ = the average words per I/O request for device n.

5. D = rhe ratio of demand to batch runs.

6. R_M = magnetic tapes requested per unit of effective time.

7. R_R = the rate at which runs are initiated expressed as runs per SUP.

4.3 COMPUTER CONFIGURATION

The model definition of the configuration consists of the following:

1. M = amount of main memory available to the user.

2. N_c = the number of CPU's.

3. $N_I(n)$ = the number of I/O requests for device type n that may be processed simultaneously.

4. $R_A(n)$ = the average access time for device n.

5. $R_T(n)$ = the transfer rate for device n.

6. L_B = the maximum batch runs allowed active simultaneously.

4.4 CPU SERVICE

For given SUP rate R_S the rate at which CPU service is requested is $R_S \cdot R_c$. The rate at which the CPU can theoretically provide service is one hour of CPU time per hour of effective time. We may use the mathematics of Section 2.0 to calculate the CPU queue time per unit of effective time as:

Q_c = CPU QUEUE RATE = QUEUE (A,B,C)

where: $A = R_S \cdot R_c$
$B = 1.$
$C = N_C$

4.5 I/O SERVICE

For SUP rate R_S and device n, the rate at which service time is requested is:

$$A = R_S \cdot R_I \cdot P(n) \ \left(R_T(n) \ \right) + \frac{R_A(n)}{\overline{W}(n)}$$

As above, with B = 1. and C = $N_I(n)$,

$Q_I(n)$ = QUEUE RATE FOR DEVICE n = QUEUE (A,B,C)

4.6 MAIN MEMORY SERVICE

Before programs can be considered for CPU and I/O services, they must be resident in the main memory of the computer. The amount of memory required is equal to the program size and varies greatly from one task to the next. The time during which the memory allocation is required is estimated by the SUP total plus the CPU and I/O queue times.

Tasks do not normally receive a single block of memory residence time. Runs are removed from main memory and swapped for others based on a complicated priority scheme. A single task may be swapped several times before it completes.

We wish to estimate the amount of time that a task seeks but is unable to receive main memory. This is done by defining the main memory as a service center and calculating the queue time from the techniques in Section 2.0. The queue time so calculated is the total wait time for memory including the hiatus prior to initial load and the portion of the swap-out periods that are due to memory competition.

To calculate the memory queue, we must define the parameters A, B, and C from Section 2.0. Recall that A is the service center input rate and B is the service rate. C is the number of requests that can be serviced simultaneously. We have already mentioned that runs require main memory for the full SUP duration plus the CPU and I/O queue times. ie:

$$A = R_S + Q_C + \sum Q_I(n)$$

$$B = 1.$$

C, the number of servers, may be translated as the number of programs that can be fit simultaneously into the user's portion of main memory. This is clearly a function of the probability that a program of given size will need main memory.

This main memory run level parameter is estimated as:

$$C = \mathrm{MAX} \Big/ \sum_{m=1}^{\mathrm{MAX}} mH(m)$$

where MAX is the maximum user memory available.

In practice H(m) is estimated by:

$$H(m) \approx \frac{\sum \mathrm{SUP}(m)}{\overline{\mathrm{SUP}}}$$

where SUP(m) is the SUP accumulation for programs of size m and $\overline{\mathrm{SUP}}$ is the total SUP accumulation for all runs.

4.7 VOLUNTARY DELAY

Regression analysis has shown that voluntary delay time is almost exclusively due to user delays on demand runs. Regression curves have been developed to estimate the delay based on two variables, the number of batch and demand runs processed. These curves must be updated periodically.

4.8 INVOLUNTARY DELAY

Regression analysis has shown that involuntary delay time is primarily incurred while magnetic tapes are mounted. Estimates are based on the number of tape mounts requested. Estimation coefficients must be updated periodically.

4.9 BATCH DELAY TIME

The batch delay valve may be considered a service center with an input rate equal to the rate at which elapsed time accumulates for batch runs, less the batch delay rate itself. The service rate is unity and the number of servers is the number of batch runs allowed to be active simultaneously (variable L_B in Section 4.1). That is:

$$B_Q = \text{BATCH DELAY TIME} = \text{QUEUE}\ (A, 1., L_B)$$

where if D = ratio of demand to batch runs

$$D_I = \text{Involuntary delay}$$

$$D_V = \text{Voluntary delay}$$

$$Q_M = \text{Memory queue}$$

$$\text{ELAPSE} = R_S + Q_C + \sum Q_I(n) + Q_M + D_I + D_V$$

then

$$A = (\text{ELAPSE} - B_Q)\ (I-D)$$

thus

$$B_Q = \text{Queue}\ (\text{ELAPSE} - B_Q, 1, L_B)$$

is an implicit function of the form

$$f(X) = X$$

and may be solved by an iterative technique. The program implementing this model uses a Wegstein approximation to evaluate B_Q.

The memory queue for batch runs is reduced by the batch delay queue since batch runs accumulate time behind the batch delay valve instead of in the memory queue.

5.0 AN EXAMPLE

Discussing the theoretical basis for the model does not suggest the way it is used in analyzing computer performance. An example will accomplish this better than abstract arguments.

The SCC has at this time, May 1976, three U1108 configurations. U1108-01 is a multiprocessing system having two central processors and 262K words of main memory. Direct access mass storage is provided by three types of device. There are 787K words available on a high speed drum system designated as an FH432. A

lower speed drum device, FH1782, provides 8.4M words. A disc device, F8440, provides 240.8M words. There are 24 tape drives available to the system. U1108-01 supports interactive demand terminals, batch terminals, and batch processing submitted from the machine room floor.

System 1108-02 has only one processor and only 131K words of main memory. Mass storage is provided by 2.4M words of FH432 drum space and 88.1M words of a very low speed drum device called Fastrand. Twelve tape drives are available. The system is used to process batch runs submitted from the floor.

The 1108-03 configuration includes a single processor and 262K words of main memory. There are 525K words of FH432, 4.2M words of FH1782, 137.6M words of F8440, and 24 tape drives available. The 03 system processes batch runs submitted both from the floor and from remote batch terminals. There are no demand (interactive) terminals connected to this system.

For this example, we will investigate the effect of discontinuing the 02 configuration. How could the remaining equipment be best utilized?

Conceptually, the analysis must define the workload and test alternative methods of processing it. Part of the workload definition should be to assess performance of the current configurations. Thus we have a benchmarking task to determine where we are, and an experimental task to assess alternatives.

The operating system of the U1108 produces data intended for use in billing computer users. These accounting data provide an excellent workload profile. Tables A, B, and C present data for the three SCC U1108 configurations depicting a week's actual work. While these profiles are not necessarily typical of future work, they will be so construed for this illustration. The workload for U1108-01 is considered in two parts since most demand terminal work is processed between 0800 and 1600 hours, Monday through Friday. The profile of demand work is distinctly different than the batch work.

A few observations can be made from an inspection of the performance data. For example, the mass storage demands on the 02 system can be absorbed by the other two systems with a net increase of less than 5% each. The profiles of mass storage usage on the 01 and 03 systems indicate that this demand can be met without impairing operations.

The main memory profiles show that the 02 system typically has greater memory demands than the other two: the average resident program is bigger. We also note that the heavy demand terminal support during the 0800-1600 period involves small programs. We probably won't want to mix the large batch programs from the 02 system with the small demand runs on the 01.

The service requirements for all three systems can be seen in figures 3, 4, and 5 which depict the SUP rate as a function of time. It is apparent that service requirements build during the 0800-1600 hour time period for the 01 and 03 systems. We will want to provide this same response even after the work from the 02 is absorbed.

To benchmark the current configuration, the model was run using the actual workloads depicted in tables A, B, and C and the actual system configuration. The results are tabulated in tables D, E, F, and G.

U1108-01 WORKLOAD
WEEK ENDING 2 MAY 1976

	0800-1600 Mon-Fri.	Other Periods
THROUGHPUT		
CPU Hours	22.2	46.2
Executive Request Charge	21.2	17.2
SUP Accumulation	91.9	126.8
Voluntary Delay	282.2	65.4
Elapsed Time Accumulation	554.4	342.9
ACTIVITY		
Number of Runs Processed	1120.0	717.0
Average Batch Runs Active	2.2	4.3
Average Demand Runs Active	12.5	1.3
Average Total Runs Active	14.8	5.5
Average Runs Not in Main Memory	8.6	1.8
PROCESSING TIME		
Total Time Not Idle	40.0	87.4
Actual Productive Time	39.2	61.8
Effective Productive Time	37.5	61.8
System Failures	0	2.0
I/O TRAFFIC PATTERNS		
Total Words Transferred	3,683,716,352.0	4,222,021,056.0
Percent on FH432	28.2	13.8
Percent on FH1782	4.5	4.9
Percent on F8440	48.6	57.3
Percent on Mag Tape	18.7	23.9
FACILITIES USAGE		
Main Memory (Core Blocks)		
Average Available	298	314
Average Used	253	223
Percent of Time 50% Full	96	82
Percent of Time 75% Full	84	55
Percent of Time 90% Full	51	23
Percent of Time 99% Full	3	2
FH432 (Tracks)		
Average Available	0	0
Average Used	439	439
Percent of Time 50% Full	100	100
Percent of Time 75% Full	100	100
Percent of Time 90% Full	100	100
Percent of Time 99% Full	100	100

(Continued)

TABLE A

	0800-1600 Mon-Fri.	Other Periods
FY1782 (Tracks)		
Average Available	397	664
Average Used	4284	4017
Percent of Time 50% Full	100	100
Percent of Time 75% Full	98	93
Percent of Time 90% Full	63	20
Percent of Time 99% Full	14	1
F8440 (Tracks)		
Average Available	44601	34763
Average Used	89799	99637
Percent of Time 50% Full	89	75
Percent of Time 75% Full	34	12
Percent of Time 90% Full	2	2
Percent of Time 99% Full	0	0
Tape Units		
Average Available	7.9	10.2
Average Used	16.1	13.8
Percent of Time 50% Full	91.0	82.0
Percent of Time 75% Full	31.0	55.0
Percent of Time 90% Full	11.0	23.0
Percent of Time 99% Full	5.0	1.0
Tapes Mounted	1485	1976

MAIN MEMORY PROFILE

Percent of SUP Total Used by Programs Occupying:

Core Blocks		
0-10	.5	.2
10-20	3.4	1.6
20-30	38.7	13.8
30-40	10.3	6.7
40-50	9.9	5.6
50-60	15.5	10.0
60-70	10.2	33.7
70-80	7.3	13.8
80-90	1.3	2.2
90-100	.7	.1
100-110	.1	1.1
110-120	.2	1.1
120-130	.8	1.1
130-140	1.1	.6
140-150		.3
150-160		6.4

(continued)

Table A Cont.

Core Blocks (Cont.)	Other Periods
160-170	.2
170-180	.0
180-190	.5
190-200	.0
200-210	.3
210-220	.0
220-230	.7

Table A Cont.

U1108-02 WORKLOAD
WEEK ENDING 2 MAY 1976

THROUGHPUT	
CPU Hours	15.5
Executive Request Charge	3.9
SUP Accumulation	82.0
Voluntary Delay	1.5
Elapsed Time Accumulation	97.2
ACTIVITY	
Number of Runs Processed	77.0
Average Batch Runs Active	1.2
Average Demand Runs Active	0.0
Average Total Runs Active	1.2
Average Runs Not in Main Memory	.0
PROCESSING TIME	
Total Time Not Idle	111.8
Actual Productive Time	89.9
Effective Productive Time	82.7
System Failures	1.0
I/O TRAFFIC PATTERNS	
Total Words Transferred	1,752,952,368.0
Percent on FH432	10.0
Percent on Fastrand	79.4
Percent on Mag Tape	10.6
FACILITIES USAGE	
Main Memory (Core Blocks)	
Average Available	162
Average Used	134
Percent of Time 50% Full	85
Percent of Time 75% Full	84
Percent of Time 90% Full	72
Percent of Time 99% Full	0
FH432 (Tracks)	
Average Available	406
Average Used	910
Percent of Time 50% Full	100
Percent of Time 75% Full	13
Percent of Time 90% Full	0
Percent of Time 99% Full	0
Fastrand (Track)	
Average Available	37506
Average Used	11646
Percent of Time 50% Full	0
Percent of Time 75% Full	0
Percent of Time 90% Full	0
Percent of Time 99% Full	0

TABLE B

FACILITIES USAGE (Cont.)

Tape Units

Average Available	9.7
Average Used	2.3
Percent of Time 50% Full	0.0
Percent of Time 75% Full	0.0
Percent of Time 90% Full	0.0
Percent of Time 99% Full	0.0
Tapes Mounted	192.0

MAIN MEMORY PROFILE

Percent of SUP Total Used by Programs Occupying:

Core Blocks	
0-10	34.1
10-20	.2
20-30	2.1
30-40	.1
40-50	.0
50-60	7.6
60-70	14.0
70-80	.0
80-90	.0
90-100	.0
100-110	.0
110-120	.0
120-130	2.0
130-140	.0
140-150	21.2
150-160	18.8

Table B Cont.

U1108-03 WORKLOAD
WEEK ENDING 2 MAY 1976

THROUGHPUT	
CPU Hours	51.7
Executive Request Charge	25.6
SUP Accumulation	176.7
Voluntary Delay	19.2
Elapsed Time Accumulation	569.3
ACTIVITY	
Number of Runs Processed	1301.0
Average Batch Runs Active	5.7
Average Demand Runs Active	0.0
Average Total Runs Active	5.7
Average Runs Not in Main Memory	1.9
PROCESSING TIME	
Total Time Not Idle	123.0
Actual Productive Time	115.3
Effective Productive Time	99.4
System Failures	1.0
I/O TRAFFIC PATTERNS	
Total Words Transferred	6,723,282,496.0
Percent on FH432	15.6
Percent on FY1782	1.7
Percent on F8440	59.8
Percent on Mag Tape	22.9
FACILITIES USAGE	
Main Memory (Core Blocks)	
Average Available	318.0
Average Used	260.0
Percent of Time 50% Full	94.0
Percent of Time 75% Full	76.0
Percent of Time 90% Full	38.0
Percent of Time 99% Full	2.9
FH432 (Tracks)	
Average Available	0
Average Used	293
Percent of Time 50% Full	100
Percent of Time 75% Full	100
Percent of Time 90% Full	100
Percent of Time 99% Full	100
FH1782 (Tracks)	
Average Available	177
Average Used	2164
Percent of Time 50% Full	100
Percent of Time 75% Full	98
Percent of Time 90% Full	77
Percent of Time 99% Full	10

TABLE C

F8440 (Tracks)	
Average Available	25936
Average Used	50864
Percent of Time 50% Full	79
Percent of Time 75% Full	32
Percent of Time 90% Full	3
Percent of Time 99% Full	0
Tape Units	
Average Available	9.3
Average Used	14.7
Percent of Time 50% Full	75
Percent of Time 75% Full	24
Percent of Time 90% Full	7
Percent of Time 99% Full	2
Tapes Mounted	3547

MAIN MEMORY PROFILE

Percent of SUP Total Used by Program Occupying:

Core Blocks	
0-10	.2
10-20	.6
20-30	11.9
30-40	7.9
40-50	7.6
50-60	23.5
60-70	24.2
70-80	13.4
80-90	1.1
90-100	1.1
100-110	1.8
110-120	1.3
120-130	.5
130-140	.8
140-150	1.2
150-160	1.9
160-170	0.0
170-180	0.0
180-190	.3
190-200	.5
200-210	0.0
210-220	.4

Table C Cont.

SUP SPEEDOMETER U-1108-01

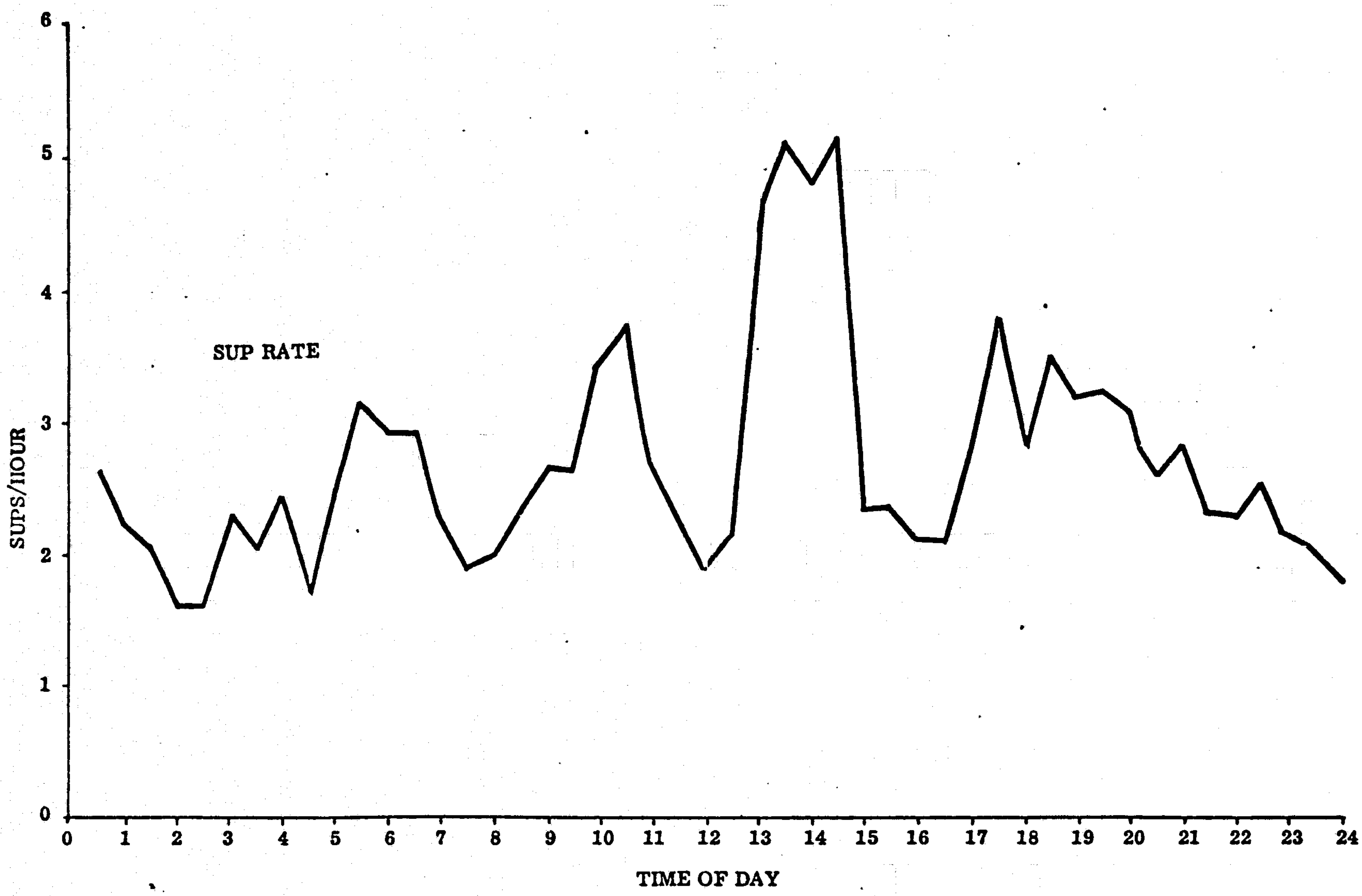

FIGURE 3

SUP SPEEDOMETER U-1108-02

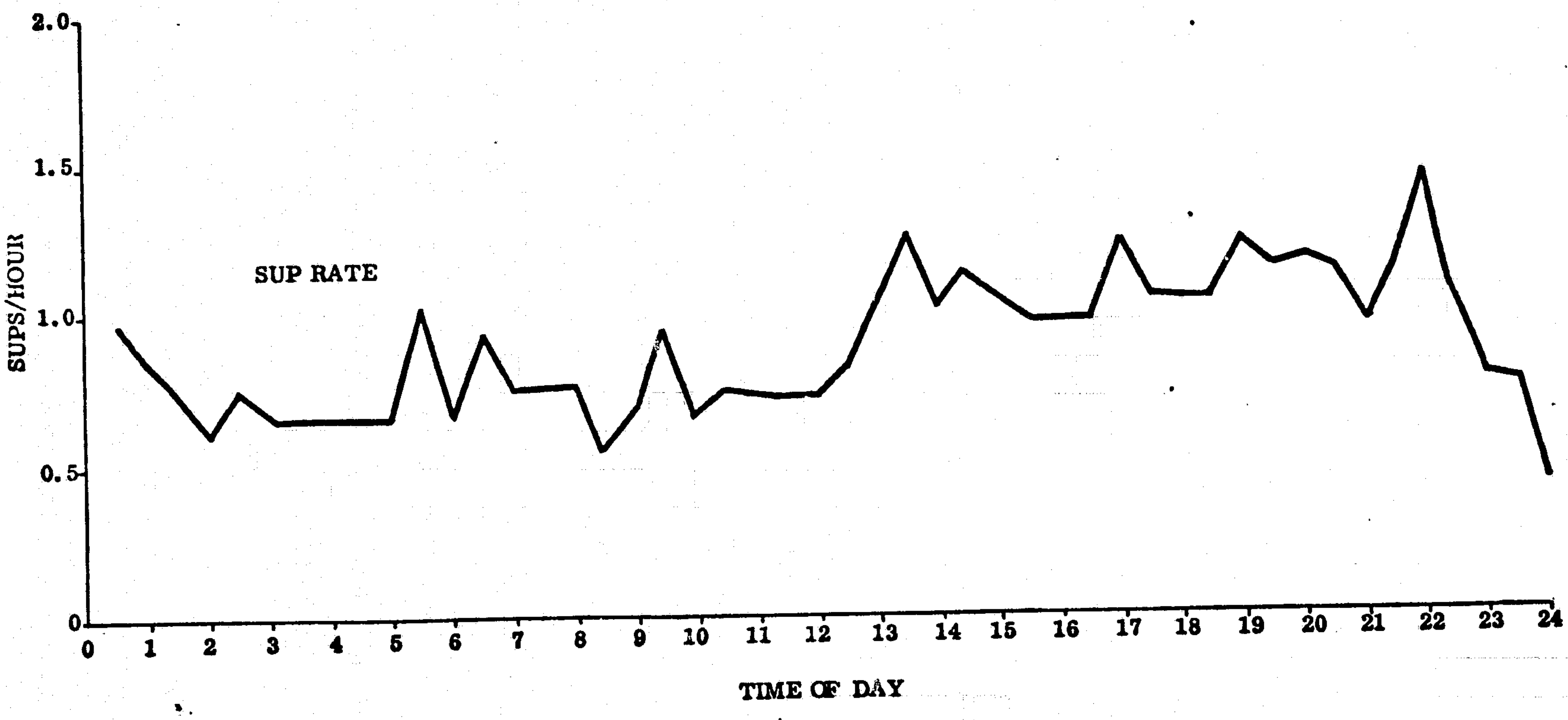

FIGURE 4

SUP SPEEDOMETER U-1108-03

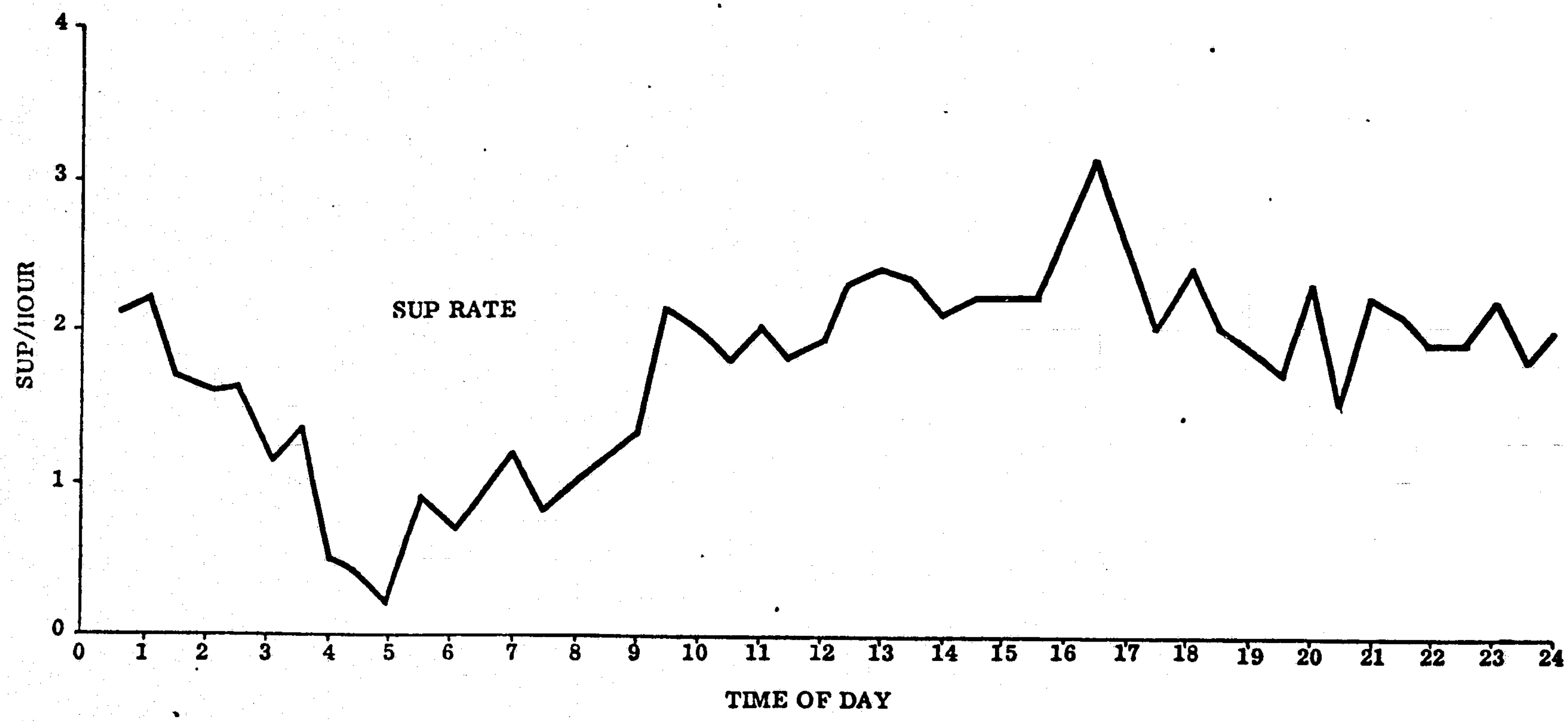

FIGURE 5

U1108-01 MODEL BENCHMARK
DAY SHIFT
WORKLOAD FROM W/E 2 MAY 1976

	Runs Per Hour	SUPS Per Hour	I/O Queue Per Hr.	CPU Queue Per Hr.	Memory Queue Per Hr.	Batch Queue Per Hr.	Voluntary Delay Per Hr.	Involuntary Delay Per Hr.	Percent Saturation
Actual Operating Level	29.8	2.45	.0169	.581	.0317	.096	7.14	2.98	71
	31.6	2.60	.0212	.741	.0591	.120	7.58	3.16	75
	32.6	2.67	.0237	.835	.0809	.134	7.80	3.25	77
	33.4	2.74	.0263	.941	.1110	.150	8.01	3.34	80
	34.3	2.82	.0292	1.050	.1540	.168	8.22	3.43	82
	35.2	2.89	.0322	1.190	.2140	.191	8.43	3.51	84
	36.1	2.96	.0354	1.340	.3020	.217	8.64	3.59	86
	36.9	3.03	.0389	1.510	.4340	.249	8.85	3.69	88
	37.8	3.10	.0425	1.700	.6390	.291	9.05	3.77	90
	38.6	3.17	.0464	1.910	.9730	.351	9.25	3.85	92
	39.5	3.24	.0504	2.160	1.5600	.441	9.45	3.94	94
	40.5	3.32	.0558	2.520	3.2400	.681	9.70	4.04	96
Saturation Level	41.7	3.42	.0626	3.040	17.2000	--	9.98	4.16	99
	42.1	3.45	.0650	3.250	74.4000	--	10.10	4.20	100

TABLE D

Table E

U1108-01 MODEL BENCHMARK
NIGHT SHIFT
WORKLOAD FROM W/E 2 MAY 1976

	Runs Per Hour	SUPS Per Hour	I/O Queue Per Hr.	CPU Queue Per Hr.	Memory Queue Per Hr.	Batch Queue Per Hr.	Voluntary Delay Per Hr.	Involuntary Delay Per Hr.	Percent Saturation
Actual Operating Level	11.6	2.05	.008	.361	.036	.924	.891	2.40	70
	12.0	2.13	.009	.415	.048	1.070	.924	2.49	72
	12.5	2.20	.010	.476	.063	1.230	.958	2.58	75
	12.9	2.28	.011	.545	.369		.991	2.68	77
	13.6	2.36	.013	.622	.497		1.020	2.77	80
	13.8	2.43	.015	.710	.678		1.060	2.86	83
	14.2	2.51	.017	.809	.944		1.090	2.95	85
	14.7	2.59	.019	.921	1.350		1.120	3.03	88
	15.1	2.66	.021	1.050	2.030		1.160	3.12	91
	15.5	2.74	.024	1.190	3.260		1.190	3.21	93
	15.9	2.81	.026	1.360	6.020		1.220	3.30	96
Saturation Level	16.4	2.89	.029	1.550	16.000		1.250	3.39	98
	16.7	2.94	.032	1.720	231.000		1.270	3.45	100

U1108-02 MODEL BENCHMARK
WORKLOAD FROM W/E 2 MAY 1976

	Runs Per Hour	SUPS Per Hour	I/O Queue Per Hr.	CPU Queue Per Hr.	Memory Queue Per Hr.	Batch Queue Per Hr.	Voluntary Delay Per Hr.	Involuntary Delay Per Hr.	Percent Saturatic
Actual Operating Level	.94	1.00	.102	.073	0	1.00	.020	.175	68
	1.01	1.08	.132	.088	.671		.021	.189	73
	1.09	1.16	.168	.104	1.050		.023	.203	78
	1.16	1.24	.211	.122	1.730		.024	.217	84
	1.24	1.32	.263	.142	3.210		.025	.231	89
	1.31	1.40	.325	.164	7.800		.027	.245	95
Saturation Level	1.39	1.48	.399	.188	131.000		.028	.259	100

Table F

U1108-03 MODEL BENCHMARK
WORKLOAD FROM W/E 2 MAY 1976

	Runs Per Hour	SUPS Per Hour	I/O Queue Per Hr.	CPU Queue Per Hr.	Memory Queue Per Hr.	Batch Queue Per Hr.	Voluntary Delay Per Hr.	Involuntary Delay Per Hr.	Percent Saturation
Actual Operating Level	11.3	1.54	.050	1.39	0	1.40	.194	2.32	85
	11.0	1.62	.058	1.72	.553		.203	2.44	89
	12.5	1.70	.068	2.15	1.38		.213	2.56	93
	13.1	1.78	.079	2.72	5.62		.223	2.68	98
Saturation Level	13.4	1.82	.085	3.09	29.9		.228	2.74	100

Table G

Looking first at the U1108-01 system and the heavy day shift workload (Table D), notice the sudden buildup in the memory queue prior to the saturation level. It is the memory queue which overloads first, causing system saturation. The CPU queue is the second most critical while the I/O queue shows capacity still available at system saturation.

Recall that CPU and I/O queue times as well as the SUP rate are included in the memory queue input rate. Therefore, we may think of these three elements as causing memory saturation. The CPU queue buildup is critical since it tends to push the memory queue into a saturation condition. Notice that the CPU queue at the actual operating level is about 1/5 of the SUP rate while at the saturation level it is nearly equal to the SUP rate. This indicates that the CPU queue is the most important contributor to the overloading of the memory queue (given the program-size profile and memory availability actually experienced).

A modeling distortion can be seen in the failure of the batch queue to saturate at the actual operating level. Since the actual batch limit was used in running the model, this queue should have saturated at the 71% level rather than the 99% level. This discrepancy is caused by the model assumption that the batch and demand work have identical profiles.

It is incorrect to assume from Table D that it would have been feasible to operate the U1108-01 system at the rate of 3.45 SUPS per hour. While this would have been theoretically possible, it would have caused an increase of over 8000% in the queue time of each run. This degradation of response time in the demand terminal environment would have been intolerable. The tradeoff of SUP rate for queue time can be seen in figure 6. It is apparent that the actual operating level is nearly optimum in terms of output gained per unit of delay. For this reason, and to be conservative, we will assume that about 70% of saturation is optimum for the day shift U1108-01.

Similarly, on the U1108-01 night shift, 70% saturation is taken as optimum. Note that the batch queue saturates closer to the actual operating level in Table E, indicating less demand influence on the total workload profile. As before, the memory queue is pushed into a saturation condition by the CPU queue (see figure 7).

The U1108-02 system seems to be running under capacity during this timeframe (see figure 8). An increase of 10% to 15% in the saturation level would effect the performance very little. It, too, is limited by the memory queue but the low speed Fastrand drums make the I/O queue more critical than on the other two systems.

The U1108-03 system appears to have been running at optimum capacity (see figure 9). Again the memory queue is pushed to saturation by the CPU queue.

From this analysis we conclude that the U1108-01 and U1108-03 systems were operated near optimum capacity during their effectively productive times in the test period.

There are several approaches to assessing the effect of removing the U1108-02 system. One way is to develop a composite workload profile from the work produced by all three systems. This profile can then be tried against optional configurations.

For example, running the composite workload against a U1108-01 configuration yields the results in table H. If we assume an optimum capacity at the 70% level, then it would be possible to produce 16.6 runs per hour. Recent studies

U-1108-01 DAY SHIFT
SUP RATE VS QUEUE RATES

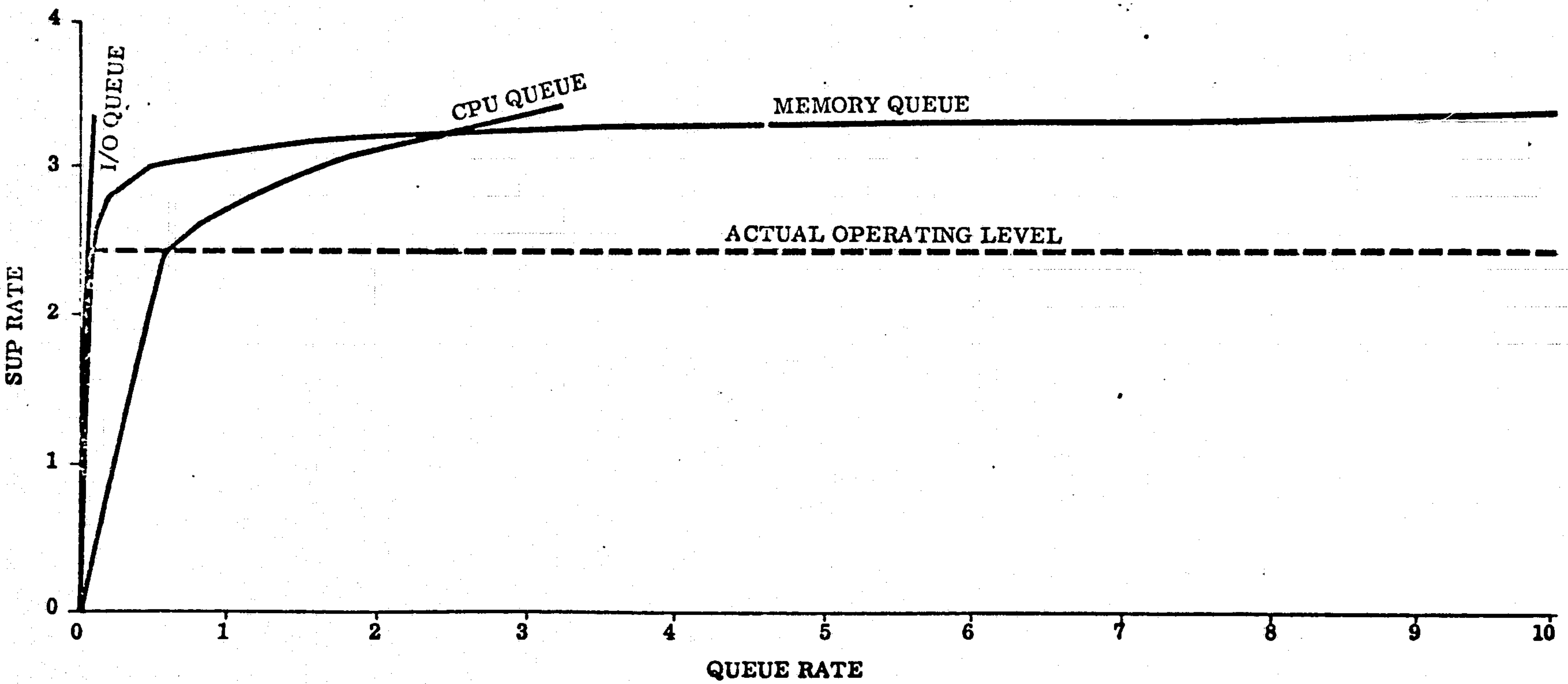

FIGURE 6

U-1108-01 NIGHT SHIFT
SUP RATE VS QUEUE RATES

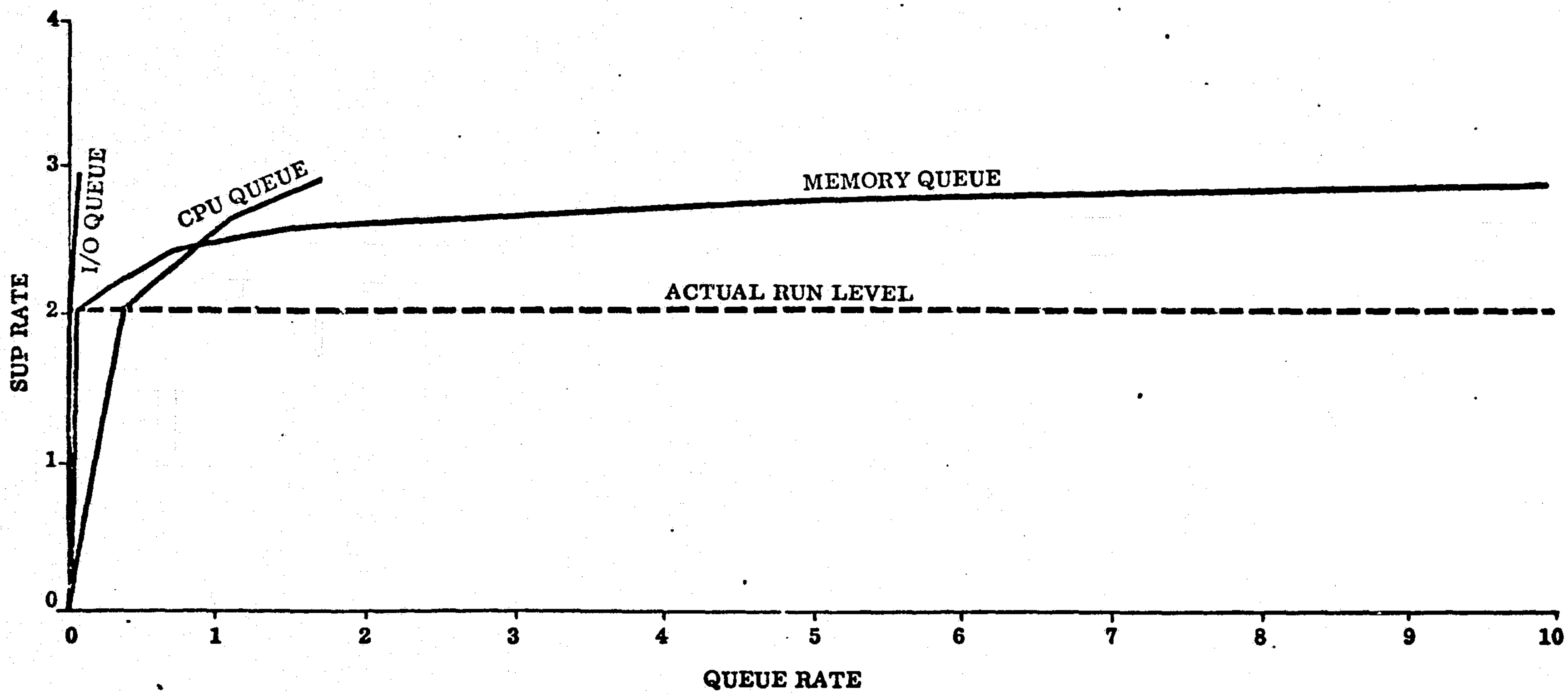

FIGURE 7

U-1108-02 SUP RATE VS QUEUE RATES

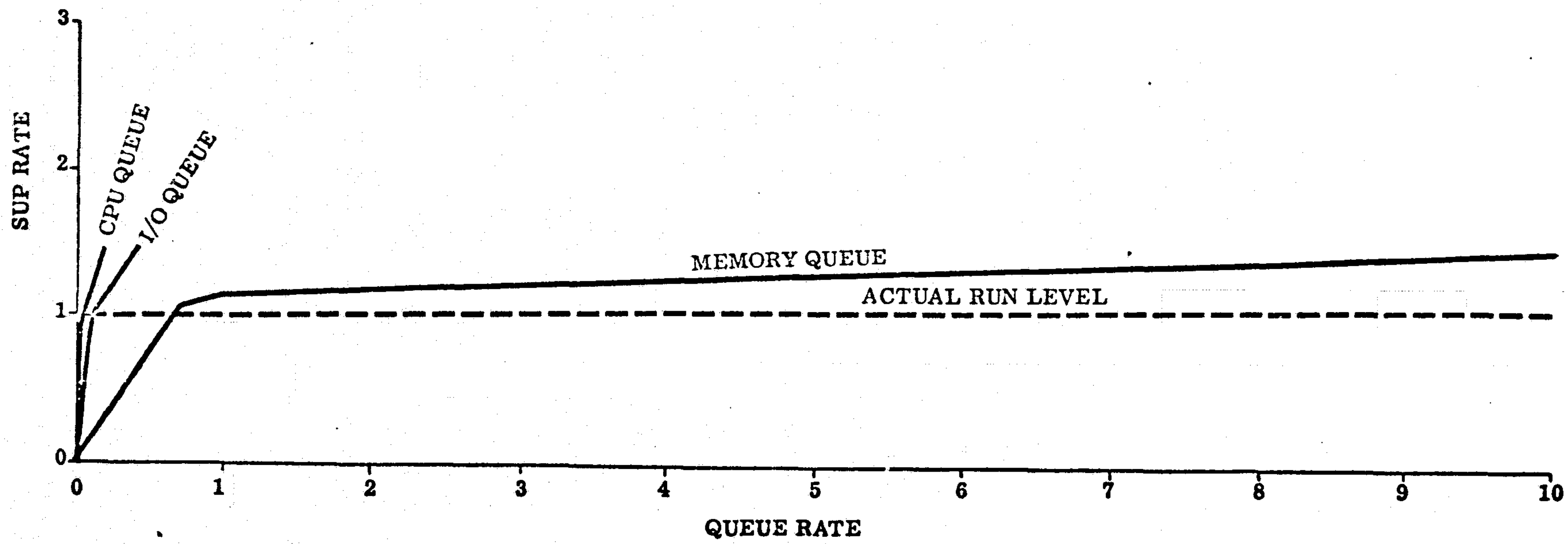

FIGURE 8

U-1108-03 SUP RATE VS QUEUE RATES

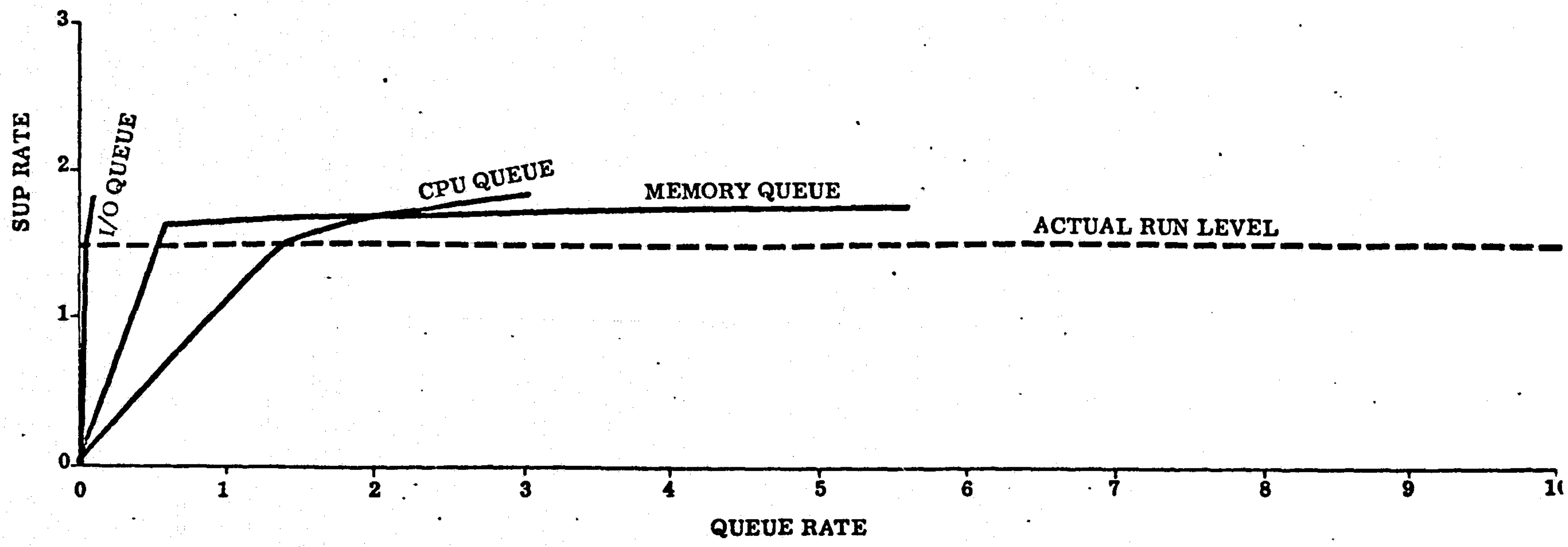

FIGURE 9

Table H

U1108-01 COMPOSITE WORKLOAD
WORKLOAD FROM W/E 2 MAY 1976

	Runs Per Hour	SUPS Per Hour	I/O Queue Per Hr.	CPU Queue Per Hr.	Memory Queue Per Hr.	Batch Queue Per Hr.	Voluntary Delay Per Hr.	Involuntary Delay Per Hr.	Percent Saturation
Expected Operating Level	16.6	2.26	.016	.399	.050	.855	1.78	2.78	71
	17.1	2.34	.018	.453	.064	.975	1.84	2.88	73
	17.7	2.42	.020	.513	.084	1.110	1.90	2.97	75
	18.2	2.49	.023	.580	.109	1.250	1.96	3.07	78
	18.8	2.57	.026	.654	.416	--	2.02	3.16	80
	19.4	2.64	.029	.738	.553	--	2.08	3.25	82
	19.9	2.72	.032	.831	.746	--	2.14	3.34	85
	20.5	2.79	.036	.936	1.030	--	2.20	3.44	87
	21.0	2.87	.040	1.050	1.460	--	2.26	3.53	89
	21.5	2.97	.044	1.180	2.160	--	2.31	3.62	92
	22.1	3.01	.049	1.330	3.450	--	2.37	3.71	94
	22.6	3.08	.053	1.500	6.330	--	2.43	3.80	96
	23.1	3.16	.058	1.690	16.500	--	2.48	3.89	98
Saturation Level	23.5	3.21	.063	1.850	148.000	--	2.53	3.95	100

indicate that effective productive time is about 85% of non-idle time (allowing for downtime and PM). There were 3215 total runs produced in the test period. At 16.6 runs per hour and 6.8 effective hours per shift, 28.5 shifts would be needed to perform the work. Two U1108-01 configurations operating 15 shifts per week could accomplish the work of the test period.

Even if the U1108-01 machine were able to reach its theoretical maximum of 23.5 runs per hour, it would require over 20 shifts of operation to complete the work. Thus, we may conclude that two U1108-01 configurations could have handled the work but one could not.

The model results of running the composite workload on the U1108-03 system are depicted in table I. If we set the expected operating level at the 85% of saturation point, as seen in the benchmark, then we would expect to produce about 10.8 runs per hour. Reasoning as for the U1108-01 we would conclude that 44 shifts of U1108-03 operation would be required by the test workload. This equates to about three such machines operating all day five days per week.

We may also conclude that together the U1108-01 and U1108-03 configurations would produce about 27.4 runs per hour and that each would require about 18 shifts of operation per week to complete the 3215 runs of the test period.

5.1 EXAMPLE CONCLUSION

The most obvious options available with existing hardware if the U1108-02 system were not available are:

1. To accomplish the work with the remaining 2 systems unchanged;
2. To acquire 262K words of additional main memory and reconfigure the CPU's into three unit processor systems similar to U1108-03;
3. To reconfigure the three processors into a single, three-CPU system; and
4. To acquire another processor and configure two, dual-CPU systems similar to U1108-01.

Of these we have seen that option 1 could not have accomplished the workload of the test period without weekend work. Options 2 and 4 accomplish the work within the 15 shifts of the standard work week. To test option 3 the composite workload was tested against the U1108-01 configuration modified to include 3 processors. The expected operating level of this configuration was 21.5 runs per week. Thus, a triple CPU configuration with maximum main memory would require about 22 shifts to complete the test period work. One such system would not be adequate.

Of the two feasible options, number 2 is the cheapest to implement. The expected operating levels of the two options do not differ significantly (33.2 runs per hour for two dual processors versus 32.4 for three unit processors - well within any reasonable estimate of the model error). The big question would concern the heavy demand workload during the day shift period. How many of the unit processors would be required to handle the day shift work now accomplished by U1108-01 and would the response times be adequate?

To answer these questions, the day shift workload profile from U1108-01 was tested against the U1108-03 configuration. The expected run level turned out to be 16.6 runs per hour indicating about 10 shifts would be required to accomplish the test period load of 1120 runs. This means two of the unit processors would have to be dedicated to the U1108-01 day shift work.

U1108-03 COMPOSITE WORKLOAD
WORKLOAD FROM W/E 2 MAY 1976

	Runs Per Hour	SUPS Per Hour	I/O Queue Per Hr.	CPU Queue Per Hr.	Memory Queue Per Hr.	Batch Queue Per Hr.	Voluntary Delay Per Hr.	Involuntary Delay Per Hr.	Percent Saturation
Expected Operating Level /	10.8	1.46	.033	1.50	.085	--	1.17	1.82	85
	11.4	1.54	.039	1.88	.200	--	1.23	1.92	90
	12.0	1.62	.046	2.40	1.640	--	1.29	2.02	94
	12.6	1.70	.053	3.12	11.300	--	1.35	2.11	99
Saturation Level /	12.7	1.72	.055	3.35	48,900	--	1.37	2.14	100

Table I

memory queues combined - excluding the batch delay queue) accrued per unit of elapsed time. This will give us a feeling for the rate at which runs are delayed because of the system load. For example, if we find queue time accuing at the rate of ½ second per second of active run time, and if the operator of a demand terminal made a request every 5 seconds, then processing of his requests would be delayed an average of 2½ seconds.

The day shift workload accrued .043 seconds of delay per second of elapsed time on the dual processor and .144 seconds per second on the unit processor. Thus, we could expect response time to about triple. We get the same relative answer but a different absolute concept of the response time if we look at queue time as a quotient of total service time. The dual processor accrues about .25 seconds of delay per SUP second while the unit processor would accrue about .87 seconds per second. Again, the response time triples.

As was mentioned at the beginning of this section, it is not the intent to develop rigorously an argument for any particular reconfiguration of the SCC computers. These examples are intended for illustrative effect. A thorough analysis would require a better development of the projected workload. There is no assurance that the workload of the week ending 2 May 1976 is representative of anything to be seen in the future. We would also require a more careful definition of the hypothetical configurations.

5.2 MODEL ACCURACY

The question of model accuracy occurs at this point as we wonder about the validity of the various performance estimates cited in this section. Accuracy estimates may be made from benchmark runs.

Comparing the model estimate of the elapsed time with the actual elapsed time accrual provides an accuracy estimate. Although several months of data should be benchmarked before any conclusive statement is made, so far the model has estimated elapsed time closely (within about 10%).

The batch delay queue can also be used to determine the accuracy of the queue time estimates. We know that this queue, unlike the others, operates at the saturation level. That is, the number of batch runs active is equal to the batch run limit set by the console operator. This is true because the batch run backlog is almost never empty.

Thus, if the model is calculating queue time correctly and if the SUP is representative of service requirements, the batch delay queue should saturate at the actual operating level. As has been pointed out, this happens for the two systems that run solely batch work but does not for the U1108-01 which runs both demand and batch.

The batch delay queue does not saturate on the U1108-01 model test at the correct level because no allowance is made for the differences between the batch and demand workload profile. This principle can be used to predict the profile of the U1108-01 batch work. On the day shift, for example, an inspection of the data in Table A indicates that the batch delay queue would have saturated at the proper level if batch work had accumulated .49 hours of elapsed time per run and required about .3 SUP hours per run. These happen to be the attributes of the work processed on the U1108-01 night shift which consists mostly of batch runs, leading to the observation that the batch delay queue seems accurate.

While this demonstration is not conclusive, it suggests a means of determining model accuracy. Confidence can be gained only over a period of extended use.

A final comment having great intuitive appeal on model accuracy will be given. When the U1108-03 benchmark test was first made, prior to the test results presented in this paper, it was noticed that the batch delay queue saturated before the supposed actual operating level. The model results were consistent with a data set that had accrued approximately 85 hours more of elapsed time than had apparently been experienced in the test period. A check was made and it was found that a program bug in the data collection routine had caused an understatement of the elapsed time amounting to 83 hours. The model was right; the data was wrong.

This example is admittedly melodramatic, but interesting.

Model accuracy depends on:

1. The accuracy of the queue calculations,
2. The accuracy of the service requirement estimates, and
3. The accuracy of the model assumptions.

Of these conditions, the most questionable is the second: service requirements estimates. The SUP does not state the exact system service load. The CPU charge does not include the total processor load. It is not apparent how much of the executive request charge is CPU time and how much is I/O. Preliminary indications are that the model is highly accurate and that current methods of estimating the service requirements are close enough for practical use. Experience with the model will allow development of a better accuracy estimate.

6.0 MODEL IMPLEMENTATION

A computer program implementing the model has been written in the FORTRAN V language to operate on the Univac 1108 computer under the EXEC VIII operating system. This program estimates accumulated elapsed time and other throughput parameters for input loads up to the system saturation level. Estimates are based on a specified workload profile and configuration definition.

6.1 STRUCTURAL OVERVIEW

The program is collected as one absolute link with no overlays. There is a main program and 8 external subprograms. The calling sequence is as depicted in Figure 10. All subprograms have one entry print designated by their respective names.

6.2 FUNCTIONAL OVERVIEW

The main program reads the configuration and workload definitions from a namelist called $INPUT. All performance parameters are calculated and the output reports are written. DELAYS calculates the voluntary and involuntary delay estimates; MEMUTL calculates the memory utilization estimate; QUEUE calculates all queue time estimates; and TMSWAP is an experimental subroutine estimating the time required to swap programs in and out of main memory. WEGIT is a MATHPAC routine used for solving an implicit function by iterations. WAIT is used in calculating queue times and PHAT is part of the experimental time-to-swap code. GAMMA is another MATHPAC routine used to evaluate the Gamma or factorial function.

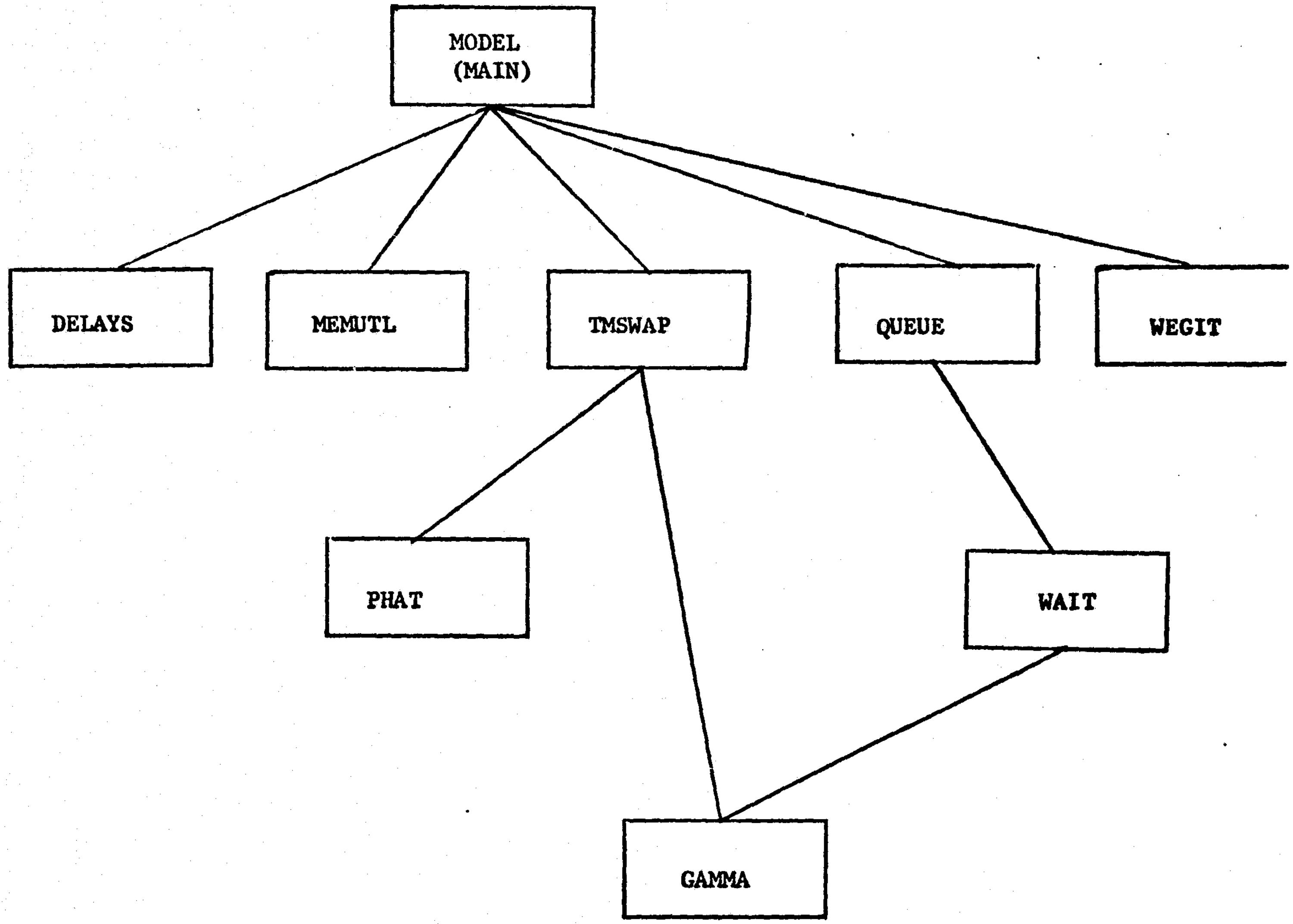
MODEL
(MAIN)
DELAYS
MEMUTL
TMSWAP
QUEUE
WEGIT
PHAT
WAIT
GAMMA

Figure 10

6.3 LOGIC FLOW AND MATHEMATICS

6.3.1 Main Program

The program reads a namelist called $INPUT. The input parameters are as depicted in table J.

The namelist is written to the standard print file for checking.

The number of words transferred is used to calculate the I/O time based on the device specifications and the I/O traffic patterns. The SUP rate is set to an initial value of .1 SUPS per hour and incremented by .02 SUPS per hour with each iteration.

In the main loop where elapsed time parameters are calculated, the input to the queue calculations is prepared. All parameters are converted to a rate per unit of effective productive time.

A call to DELAYS calculates the voluntary and involuntary delay time.

A call to TMSWAP calculates the time required for swap activity and the number of swaps per hour.

The CPU queue time is calculated by a call to QUEUE using the CPU time plus the executive request time as the input rate. This assumes that all executive request time is spent on the processor. It also assumes that these two items are exhaustive of CPU requirements. Neither assumption is entirely correct but recent system audits using SIP indicate this technique yields a reasonable estimate of CPU requirements.

The I/O queues are calculated for each device type. In this case, the input rate to the queue calculation is the time required to transfer the words indicated in the workload profile.

The memory queue is calculated using the SUP rate and the total queue rate as the input rate.

To calculate the batch delay queue, the input rate is taken as the SUP rate plus the memory queue plus voluntary and involuntary delay time less the batch queue itself. This implicit function is solved by an iterative technique using a Wegstein approximation. The input rate to the batch delay queue assumes that batch runs have the same profile as demand runs. This assumption is made in all categories of elapsed time except voluntary delay. The correct voluntary delay estimate for batch work is used. Since batch work has different service requirements than demand work, this assumption leads to some distortion of the batch delay queue when demand work is present.

The batch delay queue is subtracted from the batch portion of the memory queue since runs do not accumulate memory wait time while detained by the batch delay valve.

Output parameters are set up and written to an output file. One report is written directly to the standard output file and other parameters are written to an alternate file.

NAME	DIMENSION	TYPE	DESCRIPTION	UNITS
ACCESS	10	Real	Average access time for up to 10 device types.	seconds
XFER	10	Real	Average transfer rate for up to 10 device types.	words/sec.
MEMORY	1	Integer	Amount of user accessible main memory.	core blocks
SERV	10	Real	Number of independent I/O paths for each device type.	
NUMUNT	1	Integer	Number of I/O device types.	
NUMCPU	1	Real	Number of CPU's.	
ISWAP	1	Integer	Index of the device	
ISWAP	1	Integer	Index of the device type containing swap files.	
USEAGE	10	Real	I/O traffic patterns	Percent of words
WORDS	1	Real	Words transferred per run.	Words/run
ELR	1	Real	Elapsed time accumulated per run.	Hrs/run
CPUW	1	Real	CPU time per word	Hrs/word
ERCC	1	Real	Ratio of executive request charge to CPU time.	ERCC/CPU
VDR	1	Real	Voluntary delay per run.	Hrs/run
SIZE	1	Real	Average main memory requirements per run.	Core blocks
DEMPER	1	Real	Percent of runs that are demand runs.	
TAPR	1	Real	Tape mounts per run	Tape/run
RUNLVL	1	Real	Average limit of number of runs resident in main memory.	
BATLIM	1	Real	Maximum batch runs active.	

Table J

When the batch delay queue saturates, its value is set to zero for subsequent input levels. When any other queue saturates, the system is assumed to be saturated. A diagnostic is written and the incrementing of the SUP rate stops. The output parameters on the alternate file are written to the standard print file.

6.3.2 DELAYS (TIPMNT, BATCH, DEMAND, VOLDLL, INVLL)

This subroutine calculates:

VOLDLL: The voluntary delay estimate, and
INVLL: The involuntary delay estimate,

based on

TIPMNT: The number of tape mounts,
BATCH: The number of batch runs,
DEMAND: The number of demand runs.

Regression curves are used to calculate the two forms of delay.

6.3.3 MEMUTL (MEMSUP, SUPRAT, TOTQ)

This function calculates the memory utilization based on

MEMSUP: the SUP weighted run size,
SUPRATE: the SUP rate per hour,
TOTQ: the total queue time.

Although the calculation is trivial, it is contained in a separate subprogram because of plans to modify the model to estimate actual memory residency.

6.3.4 TMSWAP

This experimental subroutine is not yet completed.

6.3.5 PHAT

This experimental subroutine is not yet complete.

6.3.6 QUEUE (A, B, C)

This function calculates the average queue time based on the mathematics of Section 2.0. When a queue saturates, the value of QUEUE is set to -1.

The GAMMA function is used to calculate the factorial function.

6.3.8 GAMMA

A MATHPAC function.

6.3.9 WEGIT

A MATHPAC function.

6.4 INPUT

Program input comes in through one namelist (see table J). The format is as follows:

```
Card Column 1 2
               $INPUT
                ((Parameter definitions))
               $END
```

6.5 OUTPUT

Tables K, L, M, and N are the four reports output by the model.

Table K is the listing of input parameters from namelist $INPUT. (All rate parameters are expressed in terms of hours of effective productive time.)

In table L, the parameters are as follows:

SUP Rate: SUP hours per hour
RUN Rate: Runs per hour
CPU Rate: CPU hours per hour
QUEUE Rate: QUEUE hours per hour
VOLDEL Rate: Voluntary delay hours per hour as estimated by the model.
INVOL Rate: Involuntary delay hours per hour as estimated by the model.
ELAPSE Rate: Elapsed hours per hour as estimated by the model.
VOLDEL Rate (A): Actual voluntary delay hours per hour pro-rated for the run rate.
INVOL Rate (A): Actual involuntary delay hours per hour pro-rated for the run rate.
ELAPSE Rate (A): Actual elapsed hours per hour pro-rated for the run rate.
TAPMNT Delay: Involuntary delay minutes per tape mount.
BATCH QUEUE Rate: Batch queue hours per hour.

The diagnostic "QUEUE SATURATION" indicates that a queue has saturated. The following two lines indicate the values of the various queues when saturation occurred. In this case, the SWAP or memory queue saturated first and was set to -1.

The values for actual voluntary delay, involuntary delay and elapsed time are included for comparison only. This comparison is the sole purpose of inputting these parameters. They are not used in model estimates. The actual values are developed on a pro rata basis and are meaningful only in the neighborhood of the actual run level for benchmark tests. For purely hypothetical workloads, they have little or no meaning. Likewise, the minutes-per-tape-mount is valid only in the actual run level neighborhood since it is calculated from actual involuntary delay.

In table M, the parameters are as follows:

SUP Rate: same as above.
RUN Rate: same as above.
TOTAL Queue: same as above.
CPU Queue: CPU queue hours per hour.
MEMORY Queue: Memory queue hours per hour

(continued)

```
@XQT     MODEL.PROG01

@ADD,P    MODEL.TEST1
 $INPUT
 ACCESS  =      .42150000E-02,   .17000000E-01,   .30000000E-01,   .45000000E-02,
                .00000000E+00,   .00000000E+00,   .00000000E+00,   .00000000E+00,
                .00000000E+00,   .00000000E+00
 XFER    =      .12195100E+06,   .12195100E+06,   .13688900E+06,   .16000000E+05,
                .00000000E+00,   .00000000E+00,   .00000000E+00,   .00000000E+00,
                .00000000E+00,   .00000000E+00
 MEMORY  =                +298
 SERV    =      .20000000E+01,   .20000000E+01,   .30000000E+01,   .40000000E+01,
                .00000000E+00,   .00000000E+00,   .00000000E+00,   .00000000E+00,
                .00000000E+00,   .00000000E+00
 NUMCNT  =                  +4
 NUMCPU  =      .20000000E+01
 ISWAP   =                  +2
 USEAGE  =      .28200000E+00,   .45000000E-01,   .48600000E+00,   .16700000E+00,
                .00000000E+00,   .00000000E+00,   .00000000E+00,   .00000000E+00,
                .00000000E+00,   .00000000E+00
 WORDS   =      .32890320E+07
 ELR     =      .49500000E+00
 CPUW    =      .60300000E-08
 ERCC    =      .95500000E+00
 VOR     =      .25200000E+00
 SIZE    =      .34300000E+02
 DEMPER  =      .85100000E+00
 TAPR    =      .13300000E+01
 RUNLVL  =      .68500000E+01
 EXEC    =      .00000000E+00
 BATLIM  =      .22000000E+01

 $END
```

Table K
41

SUP RATE	RUN RATE	CPU RATE	QUEUE RATE	VOLDEL RATE	INVOL RATE	ELAPSE RATE	VOLDEL RATE(A)	INVOL RATE(A)	ELAPSE RATE(A)	TAP MNT DELAY	BATCH QUEUE
3.030	36.944	.733	1.982	8.847	3.685	17.545	9.310	3.965	18.287	4.8	.249
3.048	37.158	.737	2.071	8.898	3.706	17.724	9.364	3.910	18.393	4.7	.258
3.065	37.371	.741	2.167	8.949	3.728	17.909	9.417	3.849	18.499	4.6	.269
3.083	37.583	.745	2.269	9.000	3.749	18.101	9.471	3.781	18.604	4.5	.279
3.100	37.795	.750	2.390	9.051	3.770	18.301	9.524	3.704	18.709	4.4	.291
3.117	38.006	.754	2.501	9.101	3.791	18.511	9.578	3.617	18.813	4.3	.304
3.135	38.217	.758	2.632	9.152	3.812	18.731	9.631	3.520	18.917	4.2	.318
3.152	38.427	.762	2.776	9.202	3.833	18.963	9.684	3.410	19.021	4.0	.334
3.169	38.636	.766	2.934	9.252	3.854	19.209	9.736	3.285	19.125	3.8	.351
3.186	38.844	.770	3.109	9.302	3.875	19.472	9.789	3.144	19.228	3.7	.370
3.203	39.052	.775	3.305	9.352	3.895	19.755	9.841	2.982	19.331	3.4	.389
3.220	39.259	.779	3.525	9.401	3.916	20.062	9.893	2.795	19.433	3.2	.413
3.237	39.466	.783	3.774	9.451	3.937	20.398	9.945	2.579	19.536	2.9	.441
3.254	39.672	.787	4.060	9.500	3.957	20.771	9.997	2.326	19.637	2.6	.473
3.270	39.877	.791	4.392	9.549	3.978	21.189	10.049	2.028	19.739	2.3	.510
3.287	40.081	.795	4.782	9.598	3.998	21.665	10.100	1.671	19.840	1.9	.555
3.304	40.284	.799	5.248	9.647	4.018	22.217	10.152	1.237	19.941	1.4	.609
3.320	40.487	.803	5.617	9.695	4.039	22.871	10.203	.701	20.041	.8	.661
3.337	40.689	.807	6.529	9.743	4.059	23.668	10.254	.022	20.141	.0	.770
3.353	40.890	.811	7.446	9.792	4.079	24.670	10.304	-.863	20.241	-1.0	.689
3.370	41.091	.815	8.679	9.840	4.099	25.987	10.355	-2.064	20.340	-2.3	1.055
3.386	41.291	.819	10.439	9.887	4.119	27.832	10.405	-3.792	20.439	-4.1	1.303
3.402	41.489	.823	13.178	9.935	4.139	30.654	10.455	-6.499	20.537	-7.1	1.699
3.419	41.688	.827	20.326	9.982	4.158	37.885	10.505	-13.614	20.635	-14.7	.000
3.435	41.885	.831	32.170	10.030	4.178	49.812	10.555	-25.426	20.733	-27.4	.000
3.451	42.081	.835	77.732	10.077	4.198	95.457	10.605	-70.957	20.830	-76.1	.000

QUEUE SATURATION

SWAP	CPU		I/O		
-1.0000000	3.3547168	.0011810	.0000672	.0546797	.0002748

Table L

SUP RATE	RUN RATE	TOTAL QUEUE	CPU QUEUE	MEMORY QUEUE	I/O QUEUE	I/O 1 QUEUE	I/O 2 QUEUE	I/O 3 QUEUE	I/O 4 QUEUE	I/O 5 QUEUE
3.030	36.944	1.982058	1.508706	.434471	.038881	.000787	.000045	.037905	.000144	
3.048	37.158	2.071048	1.554038	.477239	.039771	.000801	.000046	.038777	.000148	
3.065	37.371	2.166532	1.600794	.525062	.040675	.000814	.000046	.039662	.000152	
3.083	37.583	2.269322	1.649031	.578699	.041591	.000828	.000047	.040559	.000156	
3.100	37.795	2.380380	1.698607	.639053	.042520	.000843	.000048	.041469	.000161	
3.117	38.006	2.500854	1.750184	.707209	.043462	.000857	.000049	.042391	.000165	
3.135	38.217	2.632122	1.803225	.784480	.044417	.000871	.000050	.043326	.000169	
3.152	38.427	2.775856	1.858000	.872472	.045384	.000886	.000050	.044274	.000174	
3.169	38.636	2.934108	1.914579	.973164	.046365	.000900	.000051	.045235	.000179	
3.186	38.844	3.109419	1.973040	1.089021	.047358	.000915	.000052	.046208	.000183	
3.203	39.052	3.304978	2.033462	1.223151	.048365	.000930	.000053	.047194	.000188	
3.220	39.259	3.524842	2.095930	1.379528	.049384	.000945	.000054	.048193	.000193	
3.237	39.466	3.774255	2.160534	1.563305	.050416	.000960	.000055	.049204	.000198	
3.254	39.672	4.060112	2.227368	1.781283	.051461	.000975	.000055	.050228	.000203	
3.270	39.677	4.391670	2.296532	2.042618	.052519	.000990	.000056	.051265	.000208	
3.287	40.081	4.781650	2.368134	2.359927	.053590	.001006	.000057	.052314	.000213	
3.304	40.284	5.248029	2.442285	2.751070	.054674	.001021	.000058	.053376	.000218	
3.320	40.487	5.817055	2.519107	3.242178	.055770	.001037	.000059	.054451	.000223	
3.337	40.689	6.528624	2.598726	3.873019	.056879	.001052	.000060	.055538	.000229	
3.353	40.890	7.446468	2.681278	4.707189	.058001	.001068	.000061	.056638	.000234	
3.370	41.091	8.679067	2.766908	5.853024	.059135	.001084	.000062	.057750	.000240	
3.386	41.291	10.437455	2.855769	7.523404	.060282	.001100	.000063	.058874	.000245	
3.402	41.489	13.178126	2.948027	10.165658	.061441	.001116	.000063	.060011	.000251	
3.419	41.686	20.325966	3.043856	17.219499	.062613	.001132	.000064	.061160	.000257	
3.435	41.885	32.169522	3.143444	28.962281	.063797	.001148	.000065	.062321	.000263	
3.451	42.081	77.731960	3.246593	74.419974	.064994	.001165	.000066	.063494	.000269	

Table M

SUP RATE	TIME TO SWAP	SWAP RATE	CPU UTIL	MEMORY UTIL	PERCENT SATURATION
3.030	.002453	27.426147	.366351	197.161409	.879227
3.048	.002504	27.997833	.368471	201.933357	.884303
3.065	.002556	28.576996	.370584	204.765606	.889364
3.083	.002609	29.163663	.372691	207.660536	.894410
3.100	.002662	29.757847	.374792	210.620653	.899440
3.117	.002716	30.359578	.376886	213.648628	.904455
3.135	.002770	30.968869	.378974	216.747221	.909454
3.152	.002825	31.585748	.381055	219.919378	.914437
3.169	.002881	32.210243	.383130	223.168196	.919403
3.186	.002938	32.842377	.385198	226.496960	.924354
3.203	.002995	33.482174	.387259	229.909111	.929287
3.220	.003053	34.129669	.389313	233.408302	.934204
3.237	.003111	34.784885	.391360	236.998413	.939104
3.254	.003171	35.447866	.393400	240.683531	.943987
3.270	.003231	36.118635	.395433	244.467983	.948852
3.287	.003291	36.797234	.397458	248.356377	.953700
3.304	.003353	37.483696	.399477	252.353590	.958530
3.320	.003415	38.178068	.401488	256.464813	.963341
3.337	.003478	38.880387	.403491	260.695560	.968135
3.353	.003541	39.590701	.405487	265.051689	.972910
3.370	.003606	40.309051	.407475	269.539478	.977667
3.386	.003671	41.035498	.409455	274.165558	.982404
3.402	.003736	41.770080	.411428	278.937057	.987123
3.419	.003803	42.512852	.413392	283.861591	.991822
3.435	.003870	43.263690	.415349	288.947289	.996502
3.451	.003938	44.023231	.417297	294.202816	1.001162

Table N

I/O QUEUE: I/O queue hours for all device types.
I/O iQUEUE: I/O queue hours per hour for device type i. The report is formatted for only five device types.

In table N, the parameters are as follows:

SUP Rate: same as above.
TIME TO SWAP: The time required to accomplish swapping activity (experimental).
CPU UTIL: Percent of time CPU produces billable service.
SWAP Rate: Swaps per hour (experimental).
MEMORY UTIL: Average number of core blocks required for resident, busy runs. Resident, delayed runs are excluded.
PERCENT SATURATION: The ratio of current-line SUP rate to that at saturation.

6.6 FILE ASSIGNMENTS

All input is read from the standard input file "READ$" equated to logical unit number 5 in the FORTRAN source code.

All reports are written to the standard print file PRINT$, FORTRAN logical unit 6.

Intermediate unformatted output is written to a sequential file named "25". This file is dynamically assigned to mass storage.

6.7 PROGRAM EXECUTION

Program execution is accomplished by the following setup:

```
Card Column 12
           @RUN
           @XQT
            $INPUT
             ((input parameters))
            $END
           @FIN
```

The program requires a total main memory allocation of about 12K decimal words. A typical execution requires between one and two minutes of CPU time.

7.0 PROGRAM LISTING

See Figure 11 for the program listing.

```
**** PERFORMANCE MODEL-21 MAY 1976 ****                                    DATE 052176      PAGE    1

@FOR,IS      .MODEL,.MODEL
FOR E2CA-05/21/76-11:32:19 (,0)

@ADD,P       MODEL.MODEL

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000672; DATA(0) 000617; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003    MEMUTL
    0004    DELAYS
    0005    THSWAP
    0006    QUEUE
    0007    WEGIT
    0010    NINTR$
    0011    NREW$
    0012    NRNL$
    0013    NWNL$
    0014    NWOU$
    0015    NIO2$
    0016    NWBU$
    0017    NIO1$
    0020    NWEF$
    0021    NRBU$
    0022    NSTOP$

   STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000     000354 10F      0001     000325 100L     0000     000430 110F     0001     000467 120L     0001     000040 124G
    0000     000434 130F     0000     000452 140F     0001     000125 147G     0001     000527 150L     0001     000562 160L
    0001     000211 167G     0000     000523 170F     0001     000577 180L     0000     000526 190F     0001     000610 200L
    0001     000643 210L     0000     000566 220F     0001     000666 230L     0001     000451 273G     0001     000101 30L
    0001     000501 312G     0001     000537 331G     0001     000572 347G     0001     000620 364G     0001     000115 40L
    0001     000657 403G     0001     000267 70L      0001     000306 80L      0001     000310 90L      0000 R   000140 ACCESS
    0000 R   000226 BATCH    0000 R   000213 BATLIM   0000 R   000244 BATQ     0000 R   000240 BATRAT   0000 R   000241 BATX
    0000 R   000245 BSWAPQ   0000 R   000251 CPUH     0000 R   000234 CPUQ     0000 R   000220 CPUR     0000 R   000230 CPURAT
    0000 R   000203 CPUW     0000 R   000225 DEMAND   0000 R   000207 DEMPER   0000 R   000214 EBAT     0000 R   000252 ELH
    0000 R   000246 ELPRAT   0000 R   000202 ELR      0000 R   000204 ERCC     0000 R   000212 EXEC     0000 I   000217 I
    0000     000261 INPUT    0000 R   000001 INVCL    0000 R   000000 INVOL    0000 I   000200 ISWAP    0000 I   000242 KBAT
    0000 I   000176 MEMORY   0000 R   000005 MEMRAT   0000 R   000002 MEMSUP   0003 R   000000 MEMUTL   0000 R   000004 MNTIM
    0000 I   000243 NBAT     0000 I   000256 NOPRT    0000 R   000003 NUMCPU   0000 I   000215 NUMPAG   0000 I   000177 NUMUNT
    0000 I   000255 NUMWRD   0000 I   000260 NWRD     0000 R   000006 OUTPRT   0000 R   000223 QUANT    0006 R   000000 QUEUE
    0000 R   000211 RUNLVL   0000 R   000224 RUNRAT   0000 R   000257 SATPER   0000 R   000254 SATRAT   0000 R   000114 SERV
    0000 R   000206 SIZE     0000 R   000216 SUM      0000 R   000221 SUPER    0000 R   000222 SUPRAT   0000 R   000237 SWAPQ
    0000 R   000232 SWOPP    0000 R   000127 TAPMNT   0000 R   000210 TAPR     0000 R   000233 TIMSWP   0000 R   000236 TOTQ
    0000 R   000070 TRAFIC   0000 R   000152 TRAFIK   0000 R   000102 TRAFIQ   0000 R   000235 TRAFQ    0000 R   000164 USEAGE
    0000 R   000247 UTLCPU   0000 R   000250 UTLMEM   0000 R   000253 VDH      0000 R   000205 VDR      0000 R   000231 VOLOLL
    0000 R   000201 WORDS    0000 R   000126 XFER
```

FIGURE 11

46

```
00100   C       THIS PROGRAM CALCULATES AN ELAPSED TIME PROFILE FOR
00100   C       SPECIFIED WORKLOADS AND CONFIGURATIONS.  ELAPSED TIME IS
00100   C       CATEGORIZED AS
00100   C             1. SERVICE TIME
00100   C                  I.  CPU
00100   C                 II.  I/O
00100   C             2. QUEUE TIME
00100   C                  I.  CPU QUEUE
00100   C                 II.  I/O QUEUE
00100   C                III.  MEMORY QUEUE
00100   C             3. VOLUNTARY DELAY
00100   C             4. INVOLUNTARY DELAY
00100   C
00100   C
00100   C       QUEUE TIMES ARE CALCULATED ASSUMING POISSON INPUT,
00100   C       EXPONENTIAL SERVICE, FIRST-COME-FIRST-SERVE PRIORITIES,
00100   C       AND NO DEFECTIONS FROM THE QUEUES.
00100   C
00100   C
00100   C       INPUT PARAMETERS ARE READ FROM A NAMELIST CALLED
00100   C       = $INPUT =.
00100   C       PARAMETERS ARE AS FOLLOWS.
00100   C
00100   C
00100   C       ACCESS(10):  AVERAGE ACCESS TIME FOR UP TO 10 I/O DEVICES
00100   C       XFER(10):  TRANSFER RATE FOR UP TO 10 DEVICES(WORDS/SEC).
00100   C       MEMORY:  AMOUNT OF MAIN MEMORY AVAILABE TO USERS(CORE BLOCKS).
00100   C              (INT).
00100   C       SERV:  NUMBER OF INDEPENDENT PATHS FOR EACH TYPE OF I/O
00100   C              DEVICE.
00100   C       NUMUNT:  THE NUMBER OF DIFFERENT TYPES OF I/O DEVICES(INT).
00100   C       NUMCPU:  THE NUMBER OF CPU'S CONFIGURED(REAL).
00100   C       ISWAP:  THE INDEX OF THE TYPE OF I/O DEVICE USED FOR SWAP
00100   C              FILES(INT).
00100   C       USEAGE(10):  THE PERCENT OF TOTAL DATA TRAFFIC OCCURRING
00100   C                    ON EACH TYPE OF I/O DEVICE.
00100   C       WORDS:  THE TOTAL DATA WORDS TRANSFERRED PER RUN .
00100   C       ELR:  THE ELAPSED TIME ACCUMULATED PER HOUR(USED ONLY FOR
00100   C             COMPARISON WITH THE MODEL CALCULATION OF ELAPSED TIME).
00100   C       CPUW:  THE HOURS OF CPU TIME PER DATA WORD TRANSFERRED.
00100   C       ERCC:  THE RATIO OF EXECUTIVE REQUEST CHARGES TO CPU TIME.
00100   C       VDR:  THE VOLUNTARY DELAY TIME PER RUN.
00100   C       SIZE:  THE AVERAGE PROGRAM SIZE.
00100   C       DEMPER:  THE PERCENT OF TOTAL RUNS THAT ARE DEMAND.
00100   C       TAPR:  TAPE MOUNTS PER RUN.
00100   C       RUNLVL:  THE AVERAGE MAXIMUM RESIDENT PROGRAMS.
00100   C       EXEC:  THE RATIO OF EXECUTIVE OVERHEAD TO CPU HOURS.
00100   C       BATLIM:  THE MAXIMUM BATCH RUNS ALLOWED.
00100   C
00100   C
00101          REAL INVOL,INVLL,MEMSUP,NUMCPU,MNTIM,MEMUTL,MEMRAT
00103          DIMENSION OUTPRT(50), TRAFIC(10), TRAFIQ(10), SERV(10), XFER(10),
00103         1ACCESS(10), TRAFIK(10), USEAGE(10)
00104          NAMELIST /INPUT/ ACCESS,XFER,MEMORY,SERV,NUMUNT,NUMCPU,ISWAP,USEAG
00104         1E,WORDS,ELR,CPUW,ERCC,VDR,SIZE,DEMPER,TAPR,RUNLVL,EXEC,BATLIM
00105          EBAT=.005
```

```
00106              NUMPAG=0
00107              REWIND 25
00110              READ (5,INPUT)
00113              MEMSUP=MEMORY/RUNLVL
00114              WRITE (6,INPUT)
00117              WRITE (6,10)
00121           10 FORMAT ('1     SUP          RUN          CPU       QUEUE     VOLDEL        IN
00121             1VOL     ELAPSE     VOLDEL       INVOL      ELAPSE    TAP MNT       BATCH',
00121             2/,'     RATE        RATE        RATE        RATE        RATE        RATE
00121             3   RATE       RATE(A)     RATE(A)     RATE(A)       DELAY        QUEUE',/)
00122              SUM=0.
00123              DO 20 I=1,NUMUNT
00123        C
00123        C
00123        C      CALCULATE SUP ACCUMULATION PER RUN BASED ON DATA TRAFIC.
00123        C
00126                TRAFIC(I)=(WORDS*USEAGE(I)/3600.)*(1./XFER(I)+ACCESS(I)/588.)
00127           20   SUM=SUM+TRAFIC(I)
00131              CPUR=CPUW*WORDS
00132              SUPER=SUM+CPUR*(1.+ERCC)     @    SUPS PER RUN BASED ON I/O TRAFIC
00133              SUPRAT=.1
00134              QUANT=(ACCESS(ISWAP)/1800.)+(SIZE*1024.)/(XFER(ISWAP)*3600.)
00135           30 CONTINUE
00136              NUMPAG=NUMPAG+1
00137              IF (NUMPAG.LE.50) GO TO 40
00141              NUMPAG=0
00142              WRITE (6,10)
00144           40 CONTINUE
00145              RUNRAT=SUPRAT/SUPER
00146              DO 50 I=1,NUMUNT
00151           50   TRAFIK(I)=TRAFIC(I)*RUNRAT      @  DATA TRAFIC PER HOUR OF OPERAT
00153              DEMAND=DEMPER*RUNRAT
00154              BATCH=RUNRAT-DEMAND
00155              TAPMNT=TAPR*RUNRAT
00156              CPURAT=CPUR*(1.+EXEC+ERCC)*RUNRAT      @ CPU PER HOUR+ OVERHEAD
00156        C
00156        C
00156        C      CALCULATE VOLUNTARY AND INVOLUNTARY DELAYS AND THE TIME
00156        C      REQUIRED TO ACCOMPLISH SWAPPING.
00156        C
00157              CALL DELAYS (TAPMNT,BATCH,DEMAND,VOLDLL,INVLL)
00160              CALL TMSWAP (TAPMNT,VOLDLL,DEMPER,SUPRAT,RUNLVL,QUANT,SWOPP,TIMSWP
00160             1)
00161              SUPRAT=SUPRAT+TIMSWP    @  INCLUDE THE TIME TO SWAP IN TOTAL SUPS
00162              CPUQ=QUEUE(CPURAT,1.,NUMCPU)      @ CALCULATE CPU QUEUE
00162        C     CPUQ=CPUQ*(CPUR/(CPUR*(1.+EXEC+ERCC)))  @ SCALE CPU QUEUE FOR USER
00163              IF (CPUQ.LT.0.) GO TO 120
00165              TRAFQ=0.
00166              DO 60 I=1,NUMUNT
00171                TRAFIQ(I)=QUEUE(TRAFIK(I),1.,SERV(I))     @ CALCULATE I/O QUEUE
00172                IF (TRAFIQ(I).LT.0.) GO TO 120
00174           60   TRAFQ=TRAFQ+TRAFIQ(I)
00176              TOTQ=TRAFQ+CPUQ
00177              MEMRAT=SUPRAT+TOTQ
00200              SWAPQ=QUEUE(MEMRAT,1.,RUNLVL)     @  CALCULATE MEMORY QUEUE
00201              IF (SWAPQ.LT.0.) GO TO 120
00201        C
```

```
      C
      C       CALCULATE BATCH QUEUE AND ADJUST SWAP QUEUE
      C
      C
              BATRAT=SUPRAT+TOTQ+SWAPQ+INVLL
              BATRAT=(BATRAT+(1.-DEMPER)*RUNRAT/60.)*(1.-DEMPER)
              BATX=.5*BATRAT
              KBAT=0
              NBAT=20
           70 CONTINUE
              IF ((BATRAT-BATX).LE.0.) GO TO 80
              BATQ=QUEUE(BATRAT-BATX,1.,BATLIM)
              IF (BATQ.GT.0.) GO TO 90
           80 BATQ=0.
              GO TO 100
           90 CALL WEGIT (BATX,BATQ,EBAT,KBAT,NBAT)
              IF (KBAT.EQ.1) GO TO 70
              IF (KBAT.NE.2) GO TO 80
          100 BSWAPQ=SWAPQ*(1.-DEMPER)
              SWAPQ=SWAPQ-BSWAPQ
              BSWAPQ=BSWAPQ-BATQ
              IF (BSWAPQ.LT.0.) BSWAPQ=0.
              SWAPQ=SWAPQ+BSWAPQ
      C
      C
              TOTQ=TOTQ+SWAPQ
      C
      C
      C       SET UP OUTPUT PARAMETERS
      C
              INVOL=(ELR-VDR)*RUNRAT-SUPRAT-TOTQ     @  ACTUAL INVOLUNTARY DELAY
              MNTIM=(INVOL/TAPMNT)*60.    @ TAPE MOUNT DELAY(MIN)
              ELPRAT=SUPRAT+VOLDLL+INVLL+TOTQ  @ MODEL ELAPSED TIME EST.
              UTLCPU=CPUR*RUNRAT/NUMCPU   @ CPU UTILIZATION
              UTLMEM=MEMUTL(MEMSUP,SUPRAT,TOTQ-SWAPQ)   @ MEMORY UTILIZATION
              CPUH=CPUR*RUNRAT  @ PRO RATED ACTUAL CPU TIME
              ELH=ELR*RUNRAT  @ PRO RATED ACTUAL ELAPSED TIME
              VDH=VDR*RUNRAT    @ PRO RATED ACTUAL VOLUNTARY DELAY
      C
      C
      C       WRITE OUTPUT ON PRINT FILE
      C
              WRITE (6,110) SUPRAT,RUNRAT,CPUH,TOTQ,VOLDLL,INVLL,ELPRAT,VDH,INVO
             1L,ELH,MNTIM,BATQ
          110 FORMAT (1X,10F10.?,F10.1,F10.3)
      C
      C
      C       WRITE ADDITIONAL OUTPUT ON ALTERNATE FILE
      C
              WRITE (25) SUPRAT,RUNRAT,TOTQ,CPUQ,SWAPQ,TRAFQ,(TRAFIQ(I),I=1,NUMU
             1NT),SUPRAT,TIMSWP,SWOPP,UTLCPU,UTLMEM
              SUPRAT=SUPRAT+.02              @  INCREMENT THE SUP RATE
              GO TO 30                       @  CALCULATE ANOTHER DATA POINT
          120 WRITE (6,130) SWAPQ,CPUQ,(TRAFIQ(I),I=1,NUMUNT)
          130 FORMAT (' QUEUE SATURATION',/,' SWAP            CPU
             1  I/O',/,(1X,7F10.7))
              SATRAT=SUPRAT-.02              @  RECAPTURE THE SATURATION SUP RATE
```

```
00317          C
00317          C
00317          C        WRITE OUTPUT PRESERVED ON ALTERNATE FILE
00317          C
00317          C
00320                   WRITE (6,140)
00322            140 FORMAT ('1','        SUP         RUN      TOTAL        CPU     MEMORY
00322               1       I/O       I/O 1       I/O 2       I/O 3       I/O 4       I/O 5',/,'
00322               2  RATE          RATE       QUEUE       QUEUE       QUEUE       QUEUE       QUEUE
00322               3       QUEUE       QUEUE       QUEUE       QUEUE',/)
00323                   END FILE 25
00324                   REWIND 25
00325                   NUMPAG=0
00326                   NUMWRD=NUMUNT+11
00327            150 READ (25,END=180) (OUTPRT(I),I=1,NUMWRD)
00335                   NOPRT=6+NUMUNT
00336                   NUMPAG=NUMPAG+1
00337                   IF (NUMPAG.LE.50) GO TO 160
00341                   NUMPAG=0
00342                   WRITE (6,140)
00344            160 CONTINUE
00345                   WRITE (6,170) (OUTPRT(I),I=1,NOPRT)
00353                   GO TO 150
00354            170 FORMAT (1X,2F10.3,9F10.6)
00355            180 REWIND 25
00356                   NUMPAG=0
00357                   WRITE (6,190)
00361            190 FORMAT ('1       SUP          TIME TO                 SWAP                CPU
00361               1          MEMORY          PERCENT',/,'        RATE             SWAP
00361               2       RATE                  UTIL             UTIL      SATURATION',/)
00362            200 READ (25,END=230) (OUTPRT(I),I=1,NUMWRD)
00370                   SATPER=OUTPRT( 1)/SATRAT
00371                   NUMPAG=NUMPAG+1
00372                   IF (NUMPAG.LE.50) GO TO 210
00374                   NUMPAG=0
00375                   WRITE (6,190)
00377            210 CONTINUE
00400                   NWRD=NUMUNT+7
00401                   WRITE (6,220) (OUTPRT(I),I=NWRD,NUMWRD),SATPER
00410                   GO TO 200
00411            220 FORMAT (1X,F10.3,(5F15.6))
00412            230 STOP
00412          C
00413                   END
END FOR
```

```
@FOR,IS     .DELAYS,.DELAYS
FOR E2CA-05/21/76-11:32:24 (,0)

@ADD,P      MODEL.DELAYS

   SUBROUTINE DELAYS     ENTRY POINT 000021

   STORAGE USED: CODE(1) 000027; DATA(0) 000011; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000005 INJP$

 00100      C        THIS SUBROUTINE CALCULATES THE INVOLUNTARY AND VOLUNTARY
 00100      C       DELAY PER HOUR OF OPERATION BASED ON:
 00100      C
 00100      C
 00100      C       VOLUNTARY DELAY:  BATCH AND DEMAND RUNS.
 00100      C       INVOLUNTARY DELAY:  NUMBER OF TAPE MOUNTS.
 00100      C
 00100      C
 00101             SUBROUTINE DELAYS (TIPMNT,BATCH,DEMAND,VOLDLL,INVLL)
 00103             REAL INVLL
 00104             INVLL=TIPMNT*4.5/60.
 00105             VOLDLL=(16.7/60.)*DEMAND+(1./60)*BATCH+(.0094/2.)
 00106             RETURN
 00106      C
 00107             END
END FOR
```

```
@FOR,IS    .MEMUTL,.MEMUTL
FOR E2CA-05/21/76-11:32:26 (,0)

@ADD,P      MODEL.MEMUTL

   FUNCTION MEMUTL       ENTRY POINT 000013

   STORAGE USED: CODE(1) 000015; DATA(0) 000005; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000001 INJP$      0000 R 000000 MEMUTL


00100        1*     C        THIS SUBROUTINE CALCULATES THE AVERAGE MEMORY REQUIRED
00100        2*     C        BY A GIVEN WORKLOAD PROFILE.
00100        3*     C
00101        4*             FUNCTION MEMUTL (MEMSUP,SUPRAT,TOTQ)
00103        5*             REAL MEMUTL,MEMSUP
00104        6*             MEMUTL=MEMSUP*(SUPRAT+TOTQ)
00105        7*             RETURN
00105        8*     C
00106        9*             END
END FOR
```

52

```
@FOR,IS    .TMSWAP,.TMSWAP
FOR E2CA-05/21/76-11:32:28 (,0)

@ADD,P      MODEL.TMSWAP

    SUBROUTINE TMSWAP    ENTRY POINT 000345

    STORAGE USED: CODE(1) 000377; DATA(0) 000244; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003    GAMMA
     0004    PHAT
     0005    ALOG10
     0006    EXP
     0007    XPRR
     0010    XPIR
     0011    XPRI
     0012    NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000100 10L      0001   000052 115G     0001   000063 124G     0001   000177 151G     0001   000263 166G
     0001   000104 20L      0001   000116 30L      0001   000122 40L      0000 R 000170 A        0000 R 000155 AN
     0000 R 000151 ANY      0000 R 000000 E        0000 I 000156 I        0000   000211 INJP$    0000 I 000002 ITER
     0000 I 000167 IY       0000 I 000001 K        0000 I 000153 N        0000 R 000173 PALPH    0000 R 000067 PB
     0000 R 000005 PD       0004 R 000000 PHAT     0000 R 000174 PHATS    0000 R 000175 PHY      0000 R 000151 PLAM
     0000 R 000162 PR       0000 R 000164 PROD     0000 R 000165 PROD1    0000 R 000163 PT       0000 R 000171 QHAT
     0000 R 000172 QINTS    0000 R 000154 SUM      0000 R 000152 SUPADJ   0000 R 000004 WAITS    0000 R 000003 WATIM
     0000 R 000157 Y        0000 R 000166 YHAT     0000 R 000160 Y1

00100      1*      C        THIS IS AN EXPERIMENTAL SUBROUTINE CALCULATING THE
00100      2*      C        TIME REQUIRED TO ACCOMPLISH SWAPPING OF PROGRAMS
00100      3*      C        IN AND OUT OF MAIN MEMORY.
00100      4*      C
00100      5*      C
00101      6*               SUBROUTINE TMSWAP (TAPMNT,VOLDEL,DEMAND,SUPRAT,RUNLVL,QUANT,SWOPP,
00101      7*              1TIMSWP,$)
00103      8*               E=.01
00104      9*               K=0
00105     10*               ITER=20
00106     11*               TIMSWP=1.
00107     12*               WATIM=1./60.
00110     13*               WAITS=TAPMNT+VOLDEL/WATIM
00111     14*               DIMENSION PD(50), PB(50)
00112     15*               PLAM=SUPRAT
00113     16*               SUPADJ=PLAM
00114     17*               DO 60 N=1,18
```

```
00117              PD(N)=0.
00120              PB(N)=0.
00121              SUM=0.
00122              AN=N
00123              DO 50 I=0,18
00126                Y=I
00127                CALL GAMMA (Y+1,Y1,$20,$10)
00130       10       Y1=ALOG10(Y1)
00131       20       CALL GAMMA (AN+Y+1,ANY,$40,$30)
00132       30       ANY=ALOG10(ANY)
00133       40       PR=(Y*ALOG10(RUNLVL))-(RUNLVL*ALOG10(EXP(1.))+Y1)
00134                PT=((AN+Y)*ALOG10(PLAM))-(PLAM*ALOG10(EXP(1.))+ANY)
00135                PR=10.**PR
00136                PT=10.**PT
00137       50       SUM=SUM+PR*PT
00141              PD(N)=DEMAND*SUM
00142              PB(N)=(1.-DEMAND)*SUM
00143       60     CONTINUE
00145            PROD=1.
00146            PROD1=PHAT(1,PD)
00147            YHAT=0.
00150            DO 70 IY=1,33
00153              Y=IY
00154              A=Y
00155              PROD=PROD*(1.-PROD1)
00156              PROD1=PHAT(A+1.,PD)
00157       70     YHAT=YHAT+Y*PROD*PROD1
00161            QHAT=QUANT*(2**(YHAT+1.)-1.)
00162            QINTS=SUPADJ/QHAT
00163            PALPH=0.
00164            PHATS=0.
00165            DO 80 N=1,18
00170              PALPH=PALPH+PD(N)*(1.-.5**N)+PB(N)*(1.-DEMAND)*(1.-.5**N)
00171              PHY=1.-(YHAT**2-71.*YHAT+1260.)/(70.*(35.-YHAT))
00172              PHATS=PHATS+PD(N)*(1.-(1.-PHY)**N)
00173       80     CONTINUE
00175            SWOPP=WAITS*PALPH+QINTS*PHATS
00176            TIMSWP=SWOPP*QUANT
00177            RETURN
00177    C
00200            END
END FOR
```

```
@FOR,IS     .PHAT,.PHAT
FOR E2CA-05/21/76-11:32:31 (,0)

@ADD,P    MODEL.PHAT

   FUNCTION PHAT         ENTRY POINT 000051

   STORAGE USED: CODE(1) 000060; DATA(C) 000023; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   XPRI
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000023 107G       0000 I 000002 I          0000   000011 INJP$     0000 R 000000 PHAT      0000 R 000001 PHY

 00101      1*             FUNCTION PHAT  (A,P)
 00103      2*             DIMENSION P(50)
 00104      3*             PHAT=0.
 00105      4*             PHY=1.-(A**2-71.*A+1260.)/(70.*(35.-A))
 00106      5*             DO 10 I=1,18
 00111      6*          10   PHAT=PHAT+P(I)*(1.-(1.-PHY)**I)
 00113      7*             RETURN
 00113      8*       C
 00114      9*             END
 END FOR
```

55

```
@FOR,IS    .QUEUE,.QUEUE
FOR E2CA-05/21/76-11:32:35 (,0)

@ADD,P     MODEL.QUEUE

   FUNCTION QUEUE        ENTRY POINT 000041

   STORAGE USED: CODE(1) 000060; DATA(C) 000007; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   WAIT
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000013 10L       0000   000003 INJP$     0000 R 000000 QUEUE     0000 R 000001 TEST     0003 R 000000 WAIT

 00100       1*      C        THIS FUNCTION CALCULATES THE AVERAGE QUEUE TIME
 00100       2*      C        FOR A SERVICE CENTER WITH POISSON INPUT, EXPONENTIAL
 00100       3*      C        SERVICE, FIRST-COME-FIRST-SERVE PRIORITIES, AND NO
 00100       4*      C        DEFECTIONS.
 00100       5*      C
 00100       6*      C
 00100       7*      C              A:  INPUT RATE
 00100       8*      C              B:  SERVICE RATE
 00100       9*      C              C:  NUMBER OF SERVERS
 00101      10*              FUNCTION QUEUE (A,B,C)
 00103      11*              TEST=B*C
 00104      12*              IF (A.LT.TEST) GO TO 10
 00106      13*              QUEUE=-1.
 00107      14*              RETURN
 00110      15*           10 CONTINUE
 00111      16*              QUEUE=(1./(B*C-A))*WAIT(A,B,C)
 00112      17*              QUEUE=QUEUE*A
 00113      18*              RETURN
 00113      19*      C
 00114      20*              END
 END FOR
```

56

```
@FOR,IS    .WAIT,.WAIT
FOR E2CA-05/21/76-11:32:38 (,0)


@ADD,P     MODEL.WAIT


@ADD,P     MODEL.MAP

   FUNCTION WAIT        ENTRY POINT 000170

   STORAGE USED: CODE(1) 000202; DATA(0) 000034; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   GAMMA
    0004   XPRR
    0005   XPRI
    0006   NWDU$
    0007   NIO2$
    0010   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000007 10L      0001   000105 124G     0001   000057 20L      0001   000065 30L      0001   000122 40L
    0001   000144 50L      0000   000011 6OF      0000 R 000002 CC       0000 R 000010 CCC      0000 R 000004 G
    0000 I 000007 I        0000 I 000003 IC       0000   000021 INJP$    0000 R 000001 RO       0000 R 000006 SUM
    0000 R 000005 TOP      0000 R 000000 WAIT


00100        C      THIS SUBROUTINE CALCULATES THE PROBABILITY OF A TASK WAITING FOR
00100        C      SERVICE GIVEN
00100        C      A=AVERAGE INPUT RATE
00100        C      B=AVERAGE SERVICE REQ
00100        C      C=NUMBER OF SERVERS
00101               FUNCTION WAIT (A,B,C)
00103               IF (B.GT.0) GO TO 10
00105               WAIT=0
00106               RETURN
00107           10  RO=A/B
00110               CC=C+1.
00111               IC=C
00112               IF (IC.LT.C) IC=IC+1
00114               IF (RO.NE.C.AND.RO.LT.1000.) GO TO 20
00116               WAIT=-1
00117               RETURN
00120           20  CALL GAMMA (CC,G,$50,$30)
00121           30  TOP=((RO**C)*C)/((C-RO)*G)
00122               SUM=TOP
```

57

```
00123                 DO 40 I=1,IC
00126                   CCC=IC-I+1
00127                   CALL GAMMA (CCC,G,$50,$40)
00130           40      SUM=SUM+(RO**(IC-I))/G
00132                 WAIT=TOP/SUM
00133                 RETURN
00134           50    CONTINUE
00135                 WRITE (6,60) G,CC,CCC,C,WAIT
00144           60    FORMAT (' GAMMA ERROR',5F15.5)
00145                 RETURN
00145         C
00146                 END
END FOR
```

```
@MAP,I     .A,MODEL.PROG01
MAP27 RL71-3 05/21/76 11:32:41 (,0)
     1.          LIB SCCLIB*MATHSTAT.
     2.          LIB SCC$*RLIB.
     3.                 IN MODEL

ADDRESS LIMITS      001000 021166       8311 IBANK WORDS DECIMAL
                    040000 046470       3385 DBANK WORDS DECIMAL
STARTING ADDRESS    020275


               SEGMENT $MAIN$          001000 021166     040000 046470

NFTV$/FOR                $(1)    001000 001022
NINFT$/FOR               $(1)    001023 002177    $(2)    040000 040034
NEFCD$/FOR                                        $(2)    040035 042250
NLOG$/SCC                $(1)    002200 002362    $(0)    042251 042365
NFTCH$/FOR               $(1)    002363 002645    $(2)    042366 042401
NCLOS$/FOR               $(1)    002646 003101    $(2)    042402 042427
NRBCK$/FOR               $(1)    003102 003124
NSWTC$/FOR               $(1)    003125 003151
NSSEL$/FOR               $(1)    003152 003212
NUPDA$/FOR               $(1)    003213 003246
NOUT$/FOR                $(1)    003247 004427    $(2)    042430 042466
NBDCV$/FOR               $(1)    004430 004555    $(2)    042467 042531
NFMT$/FOR                $(1)    004556 005440    $(2)    042532 042606
NOTIN$/FOR               $(1)    005441 005735    $(2)    042607 042612
NWBCK$/FOR               $(1)    005736 006047
NIOER$/FOR               $(1)    006050 006262    $(2)    042613 042760
NCNVT$/FOR               $(1)    006263 006504    $(2)    042761 043055
NININ$/FOR               $(1)    006505 006730    $(2)    043056 043064
NFCHK$/FOR               $(1)    006731 007720    $(2)    043065 043235
                         $(3)    007721 007721    $(4)    043236 043307
NTAB$/FOR                                         $(2)    043310 043353
OVERFL$/FOR              $(1)    007722 007731
NEXP5$/FOR               $(1)    007732 010017    $(2)    043354 043363
NEXP2$/FOR               $(1)    010020 010035    $(2)    043364 043366
NEXP6$/FOR               $(1)    010036 010232    $(2)    043367 043440
EXP$/FOR                 $(1)    010233 010322    $(2)    043441 043461
ALOG$/FOR                $(1)    010323 010442    $(2)    043462 043522
NERR$/FOR                $(1)    010443 011066    $(2)    043523 043727
NFINP$/FOR               $(1)    011067 011525    $(0)    043730 043730
                                                  $(2)    043731 044014
NWEF$/FOR                $(1)    011526 011734    $(2)    044015 044034
NFOUT$/FOR               $(1)    011735 012354    $(2)    044035 044056
NIER$/FOR                $(1)    012355 012535    $(2)    044057 044175
NCBUF$/FOR               $(1)    012536 012576
NLOUT$/FOR               $(1)    012577 013655    $(2)    044176 044233
NRWND$/FOR               $(1)    013656 013741    $(2)    044234 044245
UOMSYS(COMMONBLOCK)                                    044246 044263
H$MONITOR/FOR            $(1)    013742 015163    $(2)    044264 045035
                                                  $(4)    UOMSYS
CQQQQQ(COMMONBLOCK)                                    045036 045336
NLINP$                   $(1)    015164 016701    $(2)    045037 045221
```

```
                         $(3)    000000
GAMMA                    $(1)    016702 017071    $(0)    045222 045257
                                                  $(2)    BLANK$COMMON
WEGIT                    $(1)    017072 017267    $(0)    045260 045301
                                                  $(2)    BLANK$COMMON
WAIT                     $(1)    017270 017471    $(0)    045302 045335
                                                  $(2)    BLANK$COMMON
QUEUE                    $(1)    017472 017551    $(0)    045336 045344
                                                  $(2)    BLANK$COMMON
PHAT                     $(1)    017552 017631    $(0)    045345 045367
                                                  $(2)    BLANK$COMMON
THSWAP                   $(1)    017632 020230    $(0)    045370 045633
                                                  $(2)    BLANK$COMMON
MEMUTL                   $(1)    020231 020245    $(0)    045634 045640
                                                  $(2)    BLANK$COMMON
DELAYS                   $(1)    020246 020274    $(0)    045641 045651
                                                  $(2)    BLANK$COMMON
BLANK$COMMON(COMMONBLOCK)
MODEL                    $(1)    020275 021166    $(0)    045652 046470
                                                  $(2)    BLANK$COMMON

SYS$*RLIB$. LEVEL 71-3F
END MAP
```

60

☆ U.S. GOVERNMENT PRINTING OFFICE 1976—740-040/25 REGION NO. 4